

---

# **PubDNS Documentation**

**Mehrdad Arshad Rad**

**Aug 31, 2020**



---

## Contents

---

|          |                            |           |
|----------|----------------------------|-----------|
| <b>1</b> | <b>pubdns module</b>       | <b>3</b>  |
| <b>2</b> | <b>dns module</b>          | <b>5</b>  |
| <b>3</b> | <b>exceptions module</b>   | <b>7</b>  |
| <b>4</b> | <b>Indices and tables</b>  | <b>9</b>  |
|          | <b>Python Module Index</b> | <b>11</b> |
|          | <b>Index</b>               | <b>13</b> |



pubdns is a library for python to have more than 28K public dns servers from 190+ countries at your python script. it works based on the public-dns.info collected data and there is a wrapper based on the dnspython to resolve all type of dns records through these public dns server smoothly.

```
import pubdns
pd = pubdns.pubdns()
servers = pd.servers('US', 'los angeles')
```



# CHAPTER 1

---

## pubdns module

---

public dns python module

```
class pubdns.pubdns.PubDNS (cache_dir=None, host='https://public-dns.info/nameservers.csv',  
proxies=None, timeout=5, cache_disabled=False)
```

Bases: `object`

PubDNS class

### Parameters

- **cache\_dir** – (optional) The cache storage directory
- **host** – (optional) If you need to put public-dns.info csv data at another host.
- **proxies** – (optional) If you need to use a proxy, you can configure individual. requests with the proxies argument to any request method: `proxies = { 'http': 'http://10.10.1.10:3128', 'https': 'http://10.10.1.10:1080', }` You can also configure proxies by setting the environment variables `HTTP_PROXY` and `HTTPS_PROXY`
- **timeout** – (optional) Connection response timeout seconds
- **cache\_disabled** – (optional) Disable caching

```
data = {}
```

```
rand_server (country_id="")
```

Return a random public dns server

**Parameters** **country\_id** – two-letter ISO 3166-1 of the country, optional

**Return type** dict

```
servers (country_id, city="")
```

Return servers based on the country / city

### Parameters

- **country\_id** – two-letter ISO 3166-1 alpha-2 code of the country
- **city** – the city that the server is hosted on

**Return type** list

**set\_server** (*server*)

Add a custom dns server in memory

**Parameters** **server** – a server dict including below keys: - **country\_id**: two letter ISO 3166-1  
- **city**: city name - **name**: dns name - **server**: ip address - **reliability**: reliability number

**update** (*ttl=1440*)

Fetch and store public-dns.info dns servers information

**Parameters** **ttl** – update checks cache last modified time and updates if it expired based on the specified TTL. defaults to 1440 the unit is minute.

**xservers** (*country\_id, city=""*)

Return a generator of servers based on the country / city

**Parameters**

- **country\_id** – two-letter ISO 3166-1 alpha-2 code of the country
- **city** – the city that the server is hosted on, optional

**Return type** Generator[dict]

pubdns.pubdns.**pubdns** (*\*\*kwargs*)

Return a *PubDNS*

## CHAPTER 2

---

### dns module

---

dns module is a dnspython wrapper to query public dns servers

`pubdns.dns.resolver(servers, name, addr_types=['A'], timeout=1)`  
resolves the name through DNS servers

#### Parameters

- **servers** – list of dns servers
- **name** – domain name
- **add\_types** – dns query type list
- **timeout** – dns query timeout in second

```
import pubdns
pd = pubdns.pubdns()
servers = pd.servers('US', 'new york')
rs = pubdns.dns.resolver(servers, 'amazon.com', ['A'])
for r in rs:
    print(r)
```



## CHAPTER 3

---

### exceptions module

---

**exception** `pubdns.exceptions.UpdateError`

Bases: `Exception`

An update error occurred



## CHAPTER 4

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### p

- `pubdns.dns`, 5
- `pubdns.exceptions`, 7
- `pubdns.pubdns`, 3



## D

`data` (*pubdns.pubdns.PubDNS attribute*), 3

## P

`PubDNS` (*class in pubdns.pubdns*), 3

`pubdns()` (*in module pubdns.pubdns*), 4

`pubdns.dns` (*module*), 5

`pubdns.exceptions` (*module*), 7

`pubdns.pubdns` (*module*), 3

## R

`rand_server()` (*pubdns.pubdns.PubDNS method*), 3

`resolver()` (*in module pubdns.dns*), 5

## S

`servers()` (*pubdns.pubdns.PubDNS method*), 3

`set_server()` (*pubdns.pubdns.PubDNS method*), 4

## U

`update()` (*pubdns.pubdns.PubDNS method*), 4

`UpdateError`, 7

## X

`xservers()` (*pubdns.pubdns.PubDNS method*), 4