
tld Documentation

Release 0.7.10

Artur Barseghyan <artur.barseghyan@gmail.com>

Jun 14, 2018

Contents

1 Prerequisites	3
2 Installation	5
3 Usage examples	7
4 Update the list of TLD names	9
5 Troubleshooting	11
6 Testing	13
7 License	15
8 Support	17
9 Author	19
10 Documentation!	21
10.1 tld package	21
10.1.1 Subpackages	21
10.1.1.1 tld.commands package	21
10.1.1.1.1 Submodules	21
10.1.1.1.2 tld.commands.update_tld_names module	21
10.1.1.1.3 Module contents	22
10.1.2 Submodules	22
10.1.3 tld.conf module	22
10.1.4 tld.defaults module	22
10.1.5 tld.exceptions module	22
10.1.6 tld.helpers module	22
10.1.7 tld.test module	22
10.1.8 tld.utils module	22
10.1.9 Module contents	23
11 Indices and tables	25
Python Module Index	27

Extract the top level domain (TLD) from the URL given. List of TLD names is taken from [Mozilla](#).

Optionally raises exceptions on non-existing TLDs or silently fails (if `fail_silently` argument is set to True). Knows about active and inactive TLDs. If only active TLDs shall be matched against, `active_only` argument shall be set to True (default - False).

CHAPTER 1

Prerequisites

- Python 2.6, 2.7, 3.4, 3.5, 3.6 and PyPy

CHAPTER 2

Installation

Latest stable version on PyPI:

```
pip install tld
```

Or latest stable version from GitHub:

```
pip install https://github.com/barseghyanartur/tld/archive/stable.tar.gz
```

Or latest stable version from BitBucket:

```
pip install https://bitbucket.org/barseghyanartur/tld/get/stable.tar.gz
```


CHAPTER 3

Usage examples

Get the TLD name **as string** from the URL given:

```
from tld import get_tld

get_tld("http://www.google.co.uk")
# 'google.co.uk'

get_tld("http://www.google.idontexist", fail_silently=True)
# None
```

If you wish, you could get the TLD as **an object**:

```
from tld import get_tld

res = get_tld("http://some.subdomain.google.co.uk", as_object=True)

res
# 'google.co.uk'

res.subdomain
# 'some.subdomain'

res.domain
# 'google'

res.suffix
# 'co.uk'

res.tld
# 'google.co.uk'
```

Get TLD name, **ignoring the missing protocol**:

```
from tld import get_tld  
  
get_tld("www.google.co.uk", fix_protocol=True)  
# 'google.co.uk'
```

CHAPTER 4

Update the list of TLD names

To update-sync the tld names with the most recent version run the following from your terminal:

```
update-tld-names
```

Or simply do:

```
from tld.utils import update_tld_names  
update_tld_names()
```


CHAPTER 5

Troubleshooting

If somehow domain names listed [here](#) are not recognised, make sure you have the most recent version of TLD names in your virtual environment:

```
update-tld-names
```


CHAPTER 6

Testing

Simply type:

```
./runtests.py
```

Or use tox:

```
tox
```

Or use tox to check specific env:

```
tox -e py36
```


CHAPTER 7

License

MPL 1.1/GPL 2.0/LGPL 2.1

CHAPTER 8

Support

For any issues contact me at the e-mail given in the *Author* section.

CHAPTER 9

Author

Artur Barseghyan <artur.barseghyan@gmail.com>

CHAPTER 10

Documentation!

Contents:

10.1 tld package

10.1.1 Subpackages

10.1.1.1 tld.commands package

10.1.1.1.1 Submodules

10.1.1.1.2 tld.commands.update_tld_names module

`tld.commands.update_tld_names.main()`

Updates TLD names.

Example `python src/tld/commands/update_tld_names.py`

10.1.1.3 Module contents

10.1.2 Submodules

10.1.3 tld.conf module

10.1.4 tld.defaults module

10.1.5 tld.exceptions module

exception `tld.exceptions.TldBadUrl(url)`

Bases: `exceptions.ValueError`

`TldBadUrl.`

Supposed to be thrown when bad URL is given.

exception `tld.exceptions.TldDomainNotFound(domain_name)`

Bases: `exceptions.ValueError`

`TldDomainNotFound.`

Supposed to be thrown when domain name is not found (didn't match) the local TLD policy.

exception `tld.exceptions.TldIOError(msg=None)`

Bases: `exceptions.IOError`

`TldIOError.`

Supposed to be thrown when problems with reading/writing occur.

10.1.6 tld.helpers module

`tld.helpers.project_dir(base)`

Project dir.

`tld.helpers.PROJECT_DIR(base)`

Project dir.

10.1.7 tld.test module

`tld.utils.get_tld(url, active_only=False, fail_silently=False, as_object=False, fix_protocol=False)`

Extract the top level domain.

Extract the top level domain based on the mozilla's effective TLD names dat file. Returns a string. May throw `TldBadUrl` or `TldDomainNotFound` exceptions if there's bad URL provided or no TLD match found respectively.

Parameters

- `url (str)` – URL to get top level domain from.
- `active_only (bool)` – If set to True, only active patterns are matched.
- `fail_silently (bool)` – If set to True, no exceptions are raised and None is returned on failure.

- **as_object** (*bool*) – If set to True, `tld.utils.Result` object is returned, domain, suffix and tld properties.
- **fix_protocol** (*bool*) – If set to True, missing or wrong protocol is ignored (https is appended instead).

Returns String with top level domain (if `as_object` argument is set to False) or a `tld.utils.Result` object (if `as_object` argument is set to True); returns None on failure.

Return type str

`tld.utils.get_tld_names(fail_silently=False, retry_count=0)`
Build the tlds list if empty. Recursive.

Parameters

- **fail_silently** (*bool*) – If set to True, no exceptions are raised and None is returned on failure.
- **retry_count** (*int*) – If greater than 1, we raise an exception in order to avoid infinite loops.

Returns List of TLD names

Type iterable

`class tld.utils.Result(subdomain, domain, suffix)`
Bases: object

Container.

domain

extension

Alias of suffix.

Return str

subdomain

suffix

tld

TLD.

`tld.utils.update_tld_names(fail_silently=False)`
Update the local copy of TLDs file.

Parameters `fail_silently` (*bool*) – If set to True, no exceptions is raised on failure but boolean False returned.

Returns True on success, False on failure.

Return type bool

10.1.9 Module contents

`tld.get_tld(url, active_only=False, fail_silently=False, as_object=False, fix_protocol=False)`
Extract the top level domain.

Extract the top level domain based on the mozilla's effective TLD names dat file. Returns a string. May throw `TldBadUrl` or `TldDomainNotFound` exceptions if there's bad URL provided or no TLD match found respectively.

Parameters

- **url** (*str*) – URL to get top level domain from.
- **active_only** (*bool*) – If set to True, only active patterns are matched.
- **fail_silently** (*bool*) – If set to True, no exceptions are raised and None is returned on failure.
- **as_object** (*bool*) – If set to True, `tld.utils.Result` object is returned, `domain`, `suffix` and `tld` properties.
- **fix_protocol** (*bool*) – If set to True, missing or wrong protocol is ignored (https is appended instead).

Returns String with top level domain (if `as_object` argument is set to False) or a `tld.utils.Result` object (if `as_object` argument is set to True); returns None on failure.

Return type str

`tld.get_tld_names(fail_silently=False, retry_count=0)`

Build the tlds list if empty. Recursive.

Parameters

- **fail_silently** (*bool*) – If set to True, no exceptions are raised and None is returned on failure.
- **retry_count** (*int*) – If greater than 1, we raise an exception in order to avoid infinite loops.

Returns List of TLD names

Type iterable

`class tld.Result(subdomain, domain, suffix)`

Bases: object

Container.

`domain`

`extension`

Alias of `suffix`.

Return str

`subdomain`

`suffix`

`tld`

TLD.

`tld.update_tld_names(fail_silently=False)`

Update the local copy of TLDs file.

Parameters `fail_silently` (*bool*) – If set to True, no exceptions is raised on failure but boolean False returned.

Returns True on success, False on failure.

Return type bool

CHAPTER 11

Indices and tables

- genindex
- modindex
- search

Python Module Index

t

`tld`, 23
`tld.commands`, 22
`tld.commands.update_tld_names`, 21
`tld.conf`, 22
`tld.defaults`, 22
`tld.exceptions`, 22
`tld.helpers`, 22
`tld.utils`, 22

D

domain (tld.Result attribute), [24](#)
domain (tld.utils.Result attribute), [23](#)

E

extension (tld.Result attribute), [24](#)
extension (tld.utils.Result attribute), [23](#)

G

get_tld() (in module tld), [23](#)
get_tld() (in module tld.utils), [22](#)
get_tld_names() (in module tld), [24](#)
get_tld_names() (in module tld.utils), [23](#)

M

main() (in module tld.commands.update_tld_names), [21](#)

P

PROJECT_DIR() (in module tld.helpers), [22](#)
project_dir() (in module tld.helpers), [22](#)

R

Result (class in tld), [24](#)
Result (class in tld.utils), [23](#)

S

subdomain (tld.Result attribute), [24](#)
subdomain (tld.utils.Result attribute), [23](#)
suffix (tld.Result attribute), [24](#)
suffix (tld.utils.Result attribute), [23](#)

T

tld (module), [23](#)
tld (tld.Result attribute), [24](#)
tld (tld.utils.Result attribute), [23](#)
tld.commands (module), [22](#)
tld.commands.update_tld_names (module), [21](#)
tld.conf (module), [22](#)

tld.defaults (module), [22](#)
tld.exceptions (module), [22](#)
tld.helpers (module), [22](#)
tld.utils (module), [22](#)
TldBadUrl, [22](#)
TldDomainNotFound, [22](#)
TldIOError, [22](#)

U

update_tld_names() (in module tld), [24](#)
update_tld_names() (in module tld.utils), [23](#)