tld Documentation

Release 0.10

Artur Barseghyan <artur.barseghyan@gmail.com>

Nov 27, 2019

Contents

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).

CHAPTER 1

Prerequisites

- Python 2.7, 3.4, 3.5, 3.6, 3.7, 3.8 and PyPy

CHAPTER 2

Documentation

Documentation is available on Read the Docs.

CHAPTER 3

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 4

Usage examples

In addition to examples below, see the jupyter notebook workbook file.

4.1 Get the TLD name as string from the URL given

```
from tld import get_tld

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

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

4.2 Get the TLD as an object

```
from tld import get_tld

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

res
# 'co.uk'

res.subdomain
# 'some.subdomain'

res.domain
# 'google'

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

(continues on next page)

(continued from previous page)

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

res.parsed_url
# SplitResult(
#     scheme='http',
#     netloc='some.subdomain.google.co.uk',
#     path='',
#     query='',
#     fragment=''
# )
```

4.3 Get TLD name, ignoring the missing protocol

```
from tld import get_tld, get_fld

get_tld("www.google.co.uk", fix_protocol=True)
# 'co.uk'

get_fld("www.google.co.uk", fix_protocol=True)
# 'google.co.uk'
```

4.4 Return TLD parts as tuple

```
from tld import parse_tld

parse_tld('http://www.google.com')
# 'com', 'google', 'www'
```

4.5 Get the first level domain name as string from the URL given

```
from tld import get_fld

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

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

4.6 Check if some tld is a valid tld

```
from tld import is_tld

is_tld('co.uk)
```

(continues on next page)

(continued from previous page)

```
# True

is_tld('uk')
# True

is_tld('tld.doesnotexist')
# False

is_tld('www.google.com')
# False
```

CHAPTER 5

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 6

Custom list of TLD names

You could maintain your own custom version of the TLD names list (even multiple ones) and use them simultaneously with built in TLD names list.

You would then store them locally and provide a path to it as shown below:

```
from tld import get_tld

get_tld(
    "http://www.foreverchild",
    tld_names_local_path="tests/res/effective_tld_names_custom.dat.txt"
)
# 'foreverchild'
```

Same goes for first level domain names:

```
from tld import get_fld

get_fld(
    "http://www.foreverchild",
    tld_names_local_path="tests/res/effective_tld_names_custom.dat.txt"
)
# 'www.foreverchild'
```

Note, that in both examples shown above, there the original TLD names file has been modified in the following way:

```
...
// ===BEGIN ICANN DOMAINS===

// This one actually does not exist, added for testing purposes
foreverchild
...
```

CHAPTER 7

Free resources

To free up memory occupied by loading of custom TLD names, use `reset_tld_names` function with `tld_names_local_path` parameter.

```
from tld import get_tld, reset_tld_names

# Get TLD from a custom TLD names list
get_tld(
    "http://www.foreverchild",
    tld_names_local_path="tests/res/effective_tld_names_custom.dat.txt"
)

# Free resources occupied by the custom TLD names list
reset_tld_names("tests/res/effective_tld_names_custom.dat.txt")
```

CHAPTER 8

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 9

Testing

Simply type:

```
./runtests.py
```

Or use tox:

```
tox
```

Or use tox to check specific env:

```
tox -e py36
```

CHAPTER 10

Writing documentation

Keep the following hierarchy.

```
=====
title
=====

header
======

sub-header
----------

sub-sub-header
~~~~~~~~~~~~~~

sub-sub-sub-header
^^^^^^^^^^^^^^^^^^

sub-sub-sub-sub-header
++++++++++++++++++++++

sub-sub-sub-sub-sub-header
**************************
```

CHAPTER 11

License

MPL-1.1 OR GPL-2.0-only OR LGPL-2.1-or-later

CHAPTER 12

Support

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

CHAPTER 13

Author

Artur Barseghyan <artur.barseghyan@gmail.com>

CHAPTER 14

Docs

Contents:

14.1 Release history and notes

Sequence based identifiers are used for versioning (schema follows below):

```
major.minor[.revision]
```

- It's always safe to upgrade within the same minor version (for example, from 0.3 to 0.3.4).
- Minor version changes might be backwards incompatible. Read the release notes carefully before upgrading (for example, when upgrading from 0.3.4 to 0.4).
- All backwards incompatible changes are mentioned in this document.

14.1.1 0.10

2019-11-27

Note: This is the last release which supports Python 2.

- Make it possible to provide a custom path to the TLD names file.
- Make it possible to free up some resources occupied due to loading custom tld names by calling the `reset_tld_names` function with `tld_names_local_path` parameter.

14.1.2 0.9.8

2019-11-15

- Fix for occasional issue when some domains are not correctly recognised.

14.1.3 0.9.7

2019-10-30

Note: This release is dedicated to my newborn daughter. Happy birthday, my dear Ani.

- Handling urls that are only a TLD.
- Accepts already splitted URLs.
- Tested against Python 3.8.

14.1.4 0.9.6

2019-09-12

- Fix for update-tld-names returns a non-zero exit code on success (introduced with optimisations in 0.9.4).
- Minor tests improvements.

14.1.5 0.9.5

2019-09-11

- Tests improvements.

14.1.6 0.9.4

2019-09-11

- Optimisations in setup.py, tests and console scripts.
- Skip testing the update-tld-names functionality if no internet is available.

14.1.7 0.9.3

2019-04-05

- Added *is_tld* function.
- Docs updated.
- Upgrade test suite.

14.1.8 0.9.2

2019-01-10

- Fix an issue causing certain punycode TLDs to be deemed invalid.
- Tested against Python 3.7.

- Added tests for commands.
- Dropped Python 2.6 support.
- TLD source updated to the latest version.

14.1.9 0.9.1

2018-07-09

- Correctly handling nested TLDs.

14.1.10 0.9

2018-06-14

Note: This release contains backward incompatible changes. You should update your code.

The `active_only` option has been removed from `get_tld`, `get_fld` and `parse_url` functions. Update your code accordingly.

- Removed `active_only` option from `get_tld`, `get_fld` and `parse_url` functions.
- Correctly handling exceptions (!) in the original TLD list.
- Fixes in documentation.
- Added `parse_tld` function.
- Fixes the `python setup.py test` command.

14.1.11 0.8

2018-06-13

Note: This release contains backward incompatible changes. You should update your code.

Old `get_tld` functionality is moved to `get_fld` (first-level domain definition). The `as_object` argument (False by default) has been deprecated for `get_fld`.

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

Old behaviour

```
In: res.domain
Out: 'google'

In: res.extension
Out: 'co.uk'

In: res.subdomain
Out: 'www'

In: res.suffix
Out: 'co.uk'
```

(continues on next page)

(continued from previous page)

```
In: res.tld
Out: 'google.co.uk'
```

New behaviour

```
In: res.fld
Out: 'google.co.uk'

In: res.tld
Out: 'co.uk'

In: res.domain
Out: 'google'

In: res.subdomain
Out: 'www'
```

When used without `as_object` it returns `co.uk`.

Recap

If you have been happily using old version of `get_tld` function without `as_object` argument set to `True`, you might want to replace `get_tld` import with `get_fld` import:

```
# Old
from tld import get_tld
get_tld('http://google.co.uk')

# New
from tld import get_fld
get_fld('http://google.co.uk')
```

- Move to a Trie to match TLDs. This brings a speed up of 15-20%.
- It's now possible to search in public, private or all suffixes (old behaviour). Use `search_public` and `search_private` arguments accordingly. By default (to support old behavior), both are set to True.
- Correct TLD definitions.
- Domains like *****.xn–fiqs8s* are now recognized as well.
- Due to usage of `urlsplit` instead of `urlparse`, the initial list of TLDs is assembled quicker (a speed-up of 15-20%).
- Docs/ directory is included in source distribution tarball.
- More tests.

14.1.12 0.7.10

2018-04-07

- The `fix_protocol` argument respects protocol relative URLs.
- Change year in the license.
- Improved docstrings.

- TLD source updated to the latest version.

14.1.13 0.7.9

2017-05-02

- Added base path override for local .dat file.
- *python setup.py test* can used to execute the tests

14.1.14 0.7.8

2017-02-19

- Fix relative import in non-package for update-tls-names script. #15
- `get_tld` got a new argument `fix_protocol`, which fixes the missing protocol, having prepended "https" if missing or incorrect.

14.1.15 0.7.7

2017-02-09

- Tested against Python 3.5, 3.6 and PyPy.
- pep8 fixes.
- removed deprecated *tld.update* module. Use `update-tld-names` command instead.

14.1.16 0.7.6

2016-01-23

- Minor fixes.

14.1.17 0.7.5

2015-11-22

- Minor fixes.
- Updated tld names file to the latest version.

14.1.18 0.7.4

2015-09-24

- Exposed TLD initialization as `get_tld_names`.

14.1.19 0.7.3

2015-07-18

- Support for wheel packages.
- Fixed failure on some unicode domains.
- TLD source updated to the latest version.
- Documentation updated.

14.1.20 0.7.2

2014-09-28

- Minor fixes.

14.1.21 0.7.1

2014-09-23

- Force lower case of the URL for correct search.

14.1.22 0.7

2014-08-14

- Making it possible to obtain object instead of just extracting the TLD by setting the `as_object` argument of `get_tld` function to True.

14.1.23 0.6.4

2014-05-21

- Softened dependencies and lowered the `six` package version requirement to 1.4.0.
- Documentation improvements.

14.1.24 0.6.3

2013-12-05

- Speed up search

14.1.25 0.6.2

2013-12-03

- Fix for URLs with a port not handled correctly.
- Adding licenses.

14.1.26 0.6.1

2013-09-15

- Minor fixes.
- Credits added.

14.1.27 0.6

2013-09-12

- Fixes for Python 3 (Windows encoding).

14.1.28 0.5

2013-09-13

- Python 3 support added.

14.1.29 0.4

2013-08-03

- Tiny code improvements.
- Tests added.

14.2 tld package

14.2.1 Subpackages

14.2.1.1 tld.commands package

14.2.1.1.1 Submodules

14.2.1.1.2 tld.commands.update_tld_names module

14.2.1.1.3 Module contents

14.2.1.2 tld.sources package

14.2.1.2.1 Submodules

14.2.1.2.2 tld.sources.mozilla module

14.2.1.2.3 Module contents

14.2.2 Submodules

14.2.3 tld.base module

14.2.4 tld.bench module

14.2.5 tld.conf module

14.2.6 tld.defaults module

14.2.7 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.**TldImproperlyConfigured**(*msg=None*)
Bases: exceptions.Exception

TldImproperlyConfigured.

Supposed to be thrown when code is improperly configured. Typical use-case is when user tries to use *get_tld* function with both *search_public* and *search_private* set to False.

exception tld.exceptions.**TldIOError**(*msg=None*)
Bases: exceptions.IOError

TldIOError.

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

14.2.8 tld.helpers module

tld.helpers.**project_dir**(*base*)
Project dir.

tld.helpers.**PROJECT_DIR**(*base*)
Project dir.

14.2.9 tld.tests module

14.2.10 tld.utils module

tld.utils.**get_fld**(*url*, *fail_silently=False*, *fix_protocol=False*, *search_public=True*, *search_private=True*, *tld_names_local_path=None*, ***kwargs*)
Extract the first 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.
- **fail_silently** (*bool*) – If set to True, no exceptions are raised and None is returned on failure.
- **fix_protocol** (*bool*) – If set to True, missing or wrong protocol is ignored (https is appended instead).
- **search_public** (*bool*) – If set to True, search in public domains.
- **search_private** (*bool*) – If set to True, search in private domains.
- **tld_names_local_path** (*str*) –

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**(*url*, *fail_silently=False*, *as_object=False*, *fix_protocol=False*, *search_public=True*, *search_private=True*, *tld_names_local_path=None*)
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.

- **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).
- **search_public** (*bool*) – If set to True, search in public domains.
- **search_private** (*bool*) – If set to True, search in private domains.
- **tld_names_local_path** (*str*) –

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*, *tld_names_local_path=None*)

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.
- **tld_names_local_path** (*str*) –

Returns List of TLD names

Return type obj:*tld.utils.Trie*

`tld.utils.`**get_tld_names_container**()

Get container of all tld names.

Returns

Rtype dict

`tld.utils.`**is_tld**(*value*, *search_public=True*, *search_private=True*, *tld_names_local_path=None*)

Check if given URL is tld.

Parameters

- **value** (*str*) – URL to get top level domain from.
- **search_public** (*bool*) – If set to True, search in public domains.
- **search_private** (*bool*) – If set to True, search in private domains.
- **tld_names_local_path** (*str*) –

Returns

Return type bool

`tld.utils.`**parse_tld**(*url*, *fail_silently=False*, *fix_protocol=False*, *search_public=True*, *search_private=True*, *tld_names_local_path=None*)

Parse TLD into parts.

Parameters

- **url** –

- **fail_silently** –
- **fix_protocol** –
- **search_public** –
- **search_private** –
- **tld_names_local_path** –

Returns

Return type tuple

tld.utils.**process_url**(*url*, *fail_silently=False*, *fix_protocol=False*, *search_public=True*, *search_private=True*, *tld_names_local_path=None*)

Process URL.

Parameters

- **url** –
- **fail_silently** –
- **fix_protocol** –
- **search_public** –
- **search_private** –
- **tld_names_local_path** –

Returns

tld.utils.**reset_tld_names**(*tld_names_local_path=None*)

Reset the `tld_names` to empty value.

If `tld_names_local_path` is given, removes specified entry from `tld_names` instead.

Parameters **tld_names_local_path** (*str*) –

Returns

class tld.utils.**Result**(*tld*, *domain*, *subdomain*, *parsed_url*)

Bases: `object`

Container.

domain

extension

Alias of `tld`.

Return str

fld

First level domain.

Returns

Return type str

parsed_url

subdomain

suffix

Alias of `tld`.

Return str

tld

tld.utils.**update_tld_names**(*fail_silently=False*, *tld_names_source_url=None*, *tld_names_local_path=None*)

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.
- **tld_names_source_url** (*str*) –
- **tld_names_local_path** (*str*) –

Returns True on success, False on failure.

Return type bool

tld.utils.**update_tld_names_cli**()

CLI wrapper for update_tld_names.

Since update_tld_names returns True on success, we need to negate the result to match CLI semantics.

14.2.11 Module contents

tld.**get_fld**(*url*, *fail_silently=False*, *fix_protocol=False*, *search_public=True*, *search_private=True*, *tld_names_local_path=None*, ***kwargs*)

Extract the first 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.
- **fail_silently** (*bool*) – If set to True, no exceptions are raised and None is returned on failure.
- **fix_protocol** (*bool*) – If set to True, missing or wrong protocol is ignored (https is appended instead).
- **search_public** (*bool*) – If set to True, search in public domains.
- **search_private** (*bool*) – If set to True, search in private domains.
- **tld_names_local_path** (*str*) –

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**(*url*, *fail_silently=False*, *as_object=False*, *fix_protocol=False*, *search_public=True*, *search_private=True*, *tld_names_local_path=None*)

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.
- **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).
- **search_public** (*bool*) – If set to True, search in public domains.
- **search_private** (*bool*) – If set to True, search in private domains.
- **tld_names_local_path** (*str*) –

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*, *tld_names_local_path=None*)

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.
- **tld_names_local_path** (*str*) –

Returns List of TLD names

Return type obj:*tld.utils.Trie*

`tld.`**is_tld**(*value*, *search_public=True*, *search_private=True*, *tld_names_local_path=None*)

Check if given URL is tld.

Parameters

- **value** (*str*) – URL to get top level domain from.
- **search_public** (*bool*) – If set to True, search in public domains.
- **search_private** (*bool*) – If set to True, search in private domains.
- **tld_names_local_path** (*str*) –

Returns

Return type bool

`tld.`**parse_tld**(*url*, *fail_silently=False*, *fix_protocol=False*, *search_public=True*, *search_private=True*, *tld_names_local_path=None*)

Parse TLD into parts.

Parameters

- **url** –
- **fail_silently** –
- **fix_protocol** –
- **search_public** –

- **search_private** –
- **tld_names_local_path** –

Returns

Return type tuple

class tld.**Result**(*tld*, *domain*, *subdomain*, *parsed_url*)

Bases: object

Container.

domain

extension

Alias of tld.

Return str

fld

First level domain.

Returns

Return type str

parsed_url

subdomain

suffix

Alias of tld.

Return str

tld

tld.**update_tld_names**(*fail_silently=False*, *tld_names_source_url=None*, *tld_names_local_path=None*)

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.
- **tld_names_source_url** (*str*) –
- **tld_names_local_path** (*str*) –

Returns True on success, False on failure.

Return type bool

CHAPTER 15

Indices and tables

- genindex
- modindex
- search

Python Module Index

t

Index