
tld Documentation

Release 0.12.2

Artur Barseghyan <artur.barseghyan@gmail.com>

May 20, 2020

Contents

1	Prerequisites	3
2	Documentation	5
3	Installation	7
4	Usage examples	9
4.1	Get the TLD name as string from the URL given	9
4.2	Get the TLD as an object	9
4.3	Get TLD name, ignoring the missing protocol	10
4.4	Return TLD parts as tuple	10
4.5	Get the first level domain name as string from the URL given	10
4.6	Check if some tld is a valid tld	10
5	Update the list of TLD names	13
6	Custom TLD parsers	15
7	Custom list of TLD names	17
8	Free up resources	19
9	Support for Python 2.7 and 3.5	21
9.1	Python 2.7	21
9.2	Python 3.5	21
10	Troubleshooting	23
11	Testing	25
12	Writing documentation	27
13	License	29
14	Support	31
15	Author	33

16 Docs	35
16.1 Release history and notes	35
16.1.1 0.12.2	35
16.1.2 0.12.1	35
16.1.3 0.12	36
16.1.4 0.11.11	36
16.1.5 0.11.10	36
16.1.6 0.11.9	36
16.1.7 0.11.8	36
16.1.8 0.11.7	36
16.1.9 0.11.6	36
16.1.10 0.11.5	36
16.1.11 0.11.4	37
16.1.12 0.11.3	37
16.1.13 0.11.2	37
16.1.14 0.11.1	37
16.1.15 0.11	37
16.1.16 0.10	37
16.1.17 0.9.8	38
16.1.18 0.9.7	38
16.1.19 0.9.6	38
16.1.20 0.9.5	38
16.1.21 0.9.4	38
16.1.22 0.9.3	38
16.1.23 0.9.2	39
16.1.24 0.9.1	39
16.1.25 0.9	39
16.1.26 0.8	39
16.1.27 0.7.10	41
16.1.28 0.7.9	41
16.1.29 0.7.8	41
16.1.30 0.7.7	41
16.1.31 0.7.6	41
16.1.32 0.7.5	41
16.1.33 0.7.4	42
16.1.34 0.7.3	42
16.1.35 0.7.2	42
16.1.36 0.7.1	42
16.1.37 0.7	42
16.1.38 0.6.4	42
16.1.39 0.6.3	42
16.1.40 0.6.2	43
16.1.41 0.6.1	43
16.1.42 0.6	43
16.1.43 0.5	43
16.1.44 0.4	43
17 Indices and tables	45

Extract the top level domain (TLD) from the URL given. List of TLD names is taken from [Public Suffix](#).
Optionally raises exceptions on non-existing TLDs or silently fails (if `fail_silently` argument is set to True).

CHAPTER 1

Prerequisites

- Python 3.6, 3.7 and 3.8

Support for Python 2.7 and 3.5 is available as well.

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
```


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
```

Update the list of TLD names

To update/sync the tld names with the most recent versions run the following from your terminal:

```
update-tld-names
```

Or simply do:

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

Note, that this will update all registered TLD source parsers (not only the list of TLD names taken from Mozilla). In order to run the update for a single parser, append `uid` of that parser as argument.

```
update-tld-names mozilla
```

Custom TLD parsers

By default list of TLD names is taken from Mozilla. Parsing implemented in the `tld.utils.MozillaTLDSourceParser` class. If you want to use another parser, subclass the `tld.base.BaseTLDSourceParser`, provide `uid`, `source_url`, `local_path` and implement the `get_tld_names` method. Take the `tld.utils.MozillaTLDSourceParser` as a good example of such implementation. You could then use `get_tld` (as well as other `tld` module functions) as shown below:

```
from tld import get_tld
from some.module import CustomTLDSourceParser

get_tld(
    "http://www.google.co.uk",
    parser_class=CustomTLDSourceParser
)
```

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
from tld.utils import BaseMozillaTLDSourceParser

class CustomBaseMozillaTLDSourceParser(BaseMozillaTLDSourceParser):

    uid: str = 'custom_mozilla'
    local_path: str = 'tests/res/effective_tld_names_custom.dat.txt'

get_tld(
    "http://www.foreverchild",
    parser_class=CustomBaseMozillaTLDSourceParser
)
# 'foreverchild'
```

Same goes for first level domain names:

```
from tld import get_fld

get_fld(
    "http://www.foreverchild",
    parser_class=CustomBaseMozillaTLDSourceParser
)
# '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===
```

(continues on next page)

(continued from previous page)

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

Free up 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 parser
get_tld(
    "http://www.foreverchild",
    parser_class=CustomBaseMozillaTLDSourceParser
)

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

Support for Python 2.7 and 3.5

As you might have noticed, typing (Python 3.6+) is extensively used in the code. However, Python 3.5 will likely be supported until it's EOL. All modern recent versions (starting from *tld* 0.11.7) are fully compatible with Python 2.7 and 3.5 (just works with `pip install tld`).

Install from pip

```
pip install tld
```

Development tips follow:

9.1 Python 2.7

Install locally in development mode

```
python setup.py develop --python-tag py27
```

Prepare dist

```
./scripts/prepare_build_py27.sh
```

Run tests

```
tox -e py27
```

9.2 Python 3.5

Install locally in development mode

```
python setup.py develop --python-tag py35
```

Prepare dist

```
./scripts/prepare_build_py35.sh
```

Run tests

```
tox -e py35
```

CHAPTER 10

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
```

To update TLD names list for a single parser, specify it as an argument:

```
update-tld-names mozilla
```


CHAPTER 11

Testing

Simply type:

```
./runtests.py
```

Or use tox:

```
tox
```

Or use tox to check specific env:

```
tox -e py38
```


CHAPTER 12

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 13

License

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

CHAPTER 14

Support

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

CHAPTER 15

Author

Artur Barseghyan <artur.barseghyan@gmail.com>

Contents:

16.1 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.

16.1.1 0.12.2

2020-05-20

- Add mozilla license to dist.
- Fix MyPy issues.

16.1.2 0.12.1

2020-04-25

Note: In commemoration of [Armenian Genocide](#).

- Correctly handling domain names ending with dot(s).

16.1.3 0.12

2020-04-19

- Use Public Suffix list instead of deprecated Mozilla's MXR.

16.1.4 0.11.11

2020-03-10

- Minor speed-ups, reduce memory usage.

16.1.5 0.11.10

2020-02-05

- Python 2.7 and 3.5 fixes.

16.1.6 0.11.9

2019-12-16

- Adding test TLDs list to the package.

16.1.7 0.11.8

2019-12-13

- Minor fixes in setup.py.

16.1.8 0.11.7

2019-12-13

Note: There have been no code changes since 0.11.2. The only change is that support for Python 2.7 and 3.5 has been added.

- Added support for Python 2.7.

16.1.9 0.11.6

2019-12-12

- Targeted releases for all supported Python versions.

16.1.10 0.11.5

2019-12-12

- Targeted releases for all supported Python versions.

16.1.11 0.11.4

2019-12-12

- Changed order of the releases (Python 3.6 and up come first, then Python 3.5).
- Make all distributions except Python 3.5 universal.

16.1.12 0.11.3

2019-12-12

- Added missing resources to the Python 3.5 release.

16.1.13 0.11.2

2019-12-12

- Bring back Python 3.5 support.

16.1.14 0.11.1

2019-12-11

- Minor speed ups.
- More on adding typing.

16.1.15 0.11

2019-12-09

Note: Since introduction of parser classes, usage of `NAMES_SOURCE_URL` and `NAMES_LOCAL_PATH` of the `tld.conf` module is deprecated. Also, `tld_names_local_path` and `tld_names_source_url` arguments are deprecated as well. If you want to customise things, implement your own parser (inherit from `BaseTLDSourceParser`).

- Drop support for Python versions prior to 3.6.
- Clean-up dependencies.
- Introduce parsers.
- Drop `tld_names_source_url` and `tld_names_local_path` introduced in the previous release.
- Minor speed-ups (including tests).

16.1.16 0.10

2019-11-27

Note: This is the last release to support 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.

16.1.17 0.9.8

2019-11-15

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

16.1.18 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.

16.1.19 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.

16.1.20 0.9.5

2019-09-11

- Tests improvements.

16.1.21 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.

16.1.22 0.9.3

2019-04-05

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

16.1.23 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.

16.1.24 0.9.1

2018-07-09

- Correctly handling nested TLDs.

16.1.25 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.

16.1.26 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'

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.

16.1.27 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.

16.1.28 0.7.9

2017-05-02

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

16.1.29 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.

16.1.30 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.

16.1.31 0.7.6

2016-01-23

- Minor fixes.

16.1.32 0.7.5

2015-11-22

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

16.1.33 0.7.4

2015-09-24

- Exposed TLD initialization as `get_tld_names`.

16.1.34 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.

16.1.35 0.7.2

2014-09-28

- Minor fixes.

16.1.36 0.7.1

2014-09-23

- Force lower case of the URL for correct search.

16.1.37 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`.

16.1.38 0.6.4

2014-05-21

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

16.1.39 0.6.3

2013-12-05

- Speed up search

16.1.40 0.6.2

2013-12-03

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

16.1.41 0.6.1

2013-09-15

- Minor fixes.
- Credits added.

16.1.42 0.6

2013-09-12

- Fixes for Python 3 (Windows encoding).

16.1.43 0.5

2013-09-13

- Python 3 support added.

16.1.44 0.4

2013-08-03

- Tiny code improvements.
- Tests added.

CHAPTER 17

Indices and tables

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