
PyFunceble Documentation

Release 1.17.1. (Blue Bontebok: Caterpillar)

Nissar Chababy (@funilrys)

Apr 07, 2019

1	What can PyFuncible do?	3
2	PyFuncible in action	5
2.1	Domain testing	5
2.2	URL testing	5
2.3	Domain and URL testing with simple output	5
2.4	File content testing	5
2.5	File content testing with IDNA/Punycode conversion	5
2.6	Generated files/outputs	5
3	Dead-Hosts as place to use PyFuncible!	7
3.1	Why Dead-Hosts?	7
3.2	How do Dead-Hosts work?	7
3.3	History of Dead-Hosts	7
4	Installation	9
4.1	Requirements	9
4.1.1	Python 3.6+	9
4.1.2	colorama	9
4.1.3	domain2idna	10
4.1.4	PyYAML	10
4.1.5	requests	10
4.1.6	setuptools	10
4.1.7	urllib3	10
4.2	Get PyFuncible	10
4.2.1	Stable version	10
4.2.2	Development version	11
4.3	First steps	12
5	Update	13
5.1	Stable version	13
5.1.1	Using pip	13
5.1.2	Using the AUR (for Arch Linux users)	13
5.1.3	Pure Python method	14
5.2	Development version	14
5.2.1	For development	14
5.2.2	For usage	14

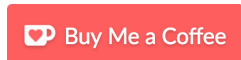
6	Configuration	17
6.1	Location	17
6.1.1	Problematics	17
6.1.2	Clone	17
6.1.3	Travis CI	17
6.1.4	Linux and MacOS (Darwin Kernel)	18
6.1.5	Windows	18
6.1.6	Custom location	19
6.2	Autoconfiguration	19
6.3	adbblock	19
6.4	auto_continue	19
6.5	command	19
6.6	command_before_end	20
6.7	custom_ip	20
6.8	days_between_db_retest	20
6.9	debug	20
6.10	filter	21
6.11	generate_hosts	21
6.12	generate_json	21
6.13	header_printed	21
6.14	hierarchical_sorting	22
6.15	iana_whois_server	22
6.16	idna_conversion	22
6.17	inactive_database	22
6.18	less	22
6.19	local	23
6.20	logs	23
6.21	mining	23
6.22	no_files	23
6.23	no_special	23
6.24	no_whois	23
6.25	plain_list_domain	24
6.26	quiet	24
6.27	referer	24
6.28	seconds_before_http_timeout	24
6.29	share_logs	24
6.30	show_execution_time	25
6.31	show_percentage	25
6.32	simple	25
6.33	split	25
6.34	syntax	25
6.35	travis	26
6.36	travis_autosave_commit	26
6.37	travis_autosave_final_commit	26
6.38	travis_autosave_minutes	26
6.39	travis_branch	26
6.40	unified	27
6.41	user_agent	27
6.42	verify_ssl_certificate	27
6.43	whois_database	27
6.44	outputs	27
6.44.1	outputs[default_files]	28
6.44.2	outputs[domains]	29
6.44.3	outputs[hosts]	29

6.44.4	outputs[json]	29
6.44.5	outputs[analytic]	30
6.44.6	outputs[logs]	32
6.44.7	outputs[main]	33
6.44.8	outputs[parent_directory]	34
6.44.9	outputs[splited]	34
6.45	status	34
6.45.1	status[list]	34
6.45.2	status[official]	36
6.46	http_codes	37
6.46.1	http_codes[active]	37
6.46.2	http_codes[list]	37
6.47	links	38
6.47.1	links[api_date_format]	39
6.47.2	links[api_no_referer]	39
6.47.3	links[config]	39
6.47.4	links[dir_structure]	39
6.47.5	links[iana]	39
6.47.6	links[repo]	39
6.47.7	links[requirements]	40
6.47.8	links[psl]	40
6.48	counter	40
6.48.1	counter[number]	40
6.48.2	counter[percentage]	41
7	Usage	43
7.1	From a terminal	43
7.1.1	Detailed	43
7.1.2	Global overview	52
7.2	From a Python script or module	55
7.2.1	Availability check of domains, IP or URL	55
7.2.2	Syntax check of domains, IP or URL	55
7.2.3	IPv4 Range and subdomain syntax check	56
7.2.4	Loop example	56
7.2.5	Advanced example	57
7.2.6	Custom Configuration	58
7.3	From Travis CI	59
7.3.1	Configuration	59
7.3.2	Getting a GitHub token	60
7.3.3	Encrypting the token for Travis CI usage	61
8	Columns	63
8.1	Domains	63
8.2	Status	63
8.2.1	ACTIVE	63
8.2.2	INACTIVE	63
8.2.3	INVALID	64
8.2.4	VALID	64
8.3	Expiration Date	64
8.3.1	Unknown	64
8.3.2	A date	64
8.4	Source	64
8.4.1	HTTP Code	64
8.4.2	IANA	65

8.4.3	SYNTAX	65
8.4.4	NSLOOKUP	65
8.4.5	SPECIAL	65
8.5	HTTP Code	66
8.5.1	As active	67
8.5.2	As potentially active	67
8.5.3	As inactive or potentially inactive	68
9	Logs Sharing	69
9.1	Why logs sharing?	69
9.2	What do we share/collect?	69
9.3	How to share logs?	69
10	Logic representation	71
10.1	Domain testing	73
10.2	IP testing	74
10.3	URL testing	76
11	API documentation	77
12	Code Documentation	81
12.1	Helpers	81
12.1.1	Problematic	81
12.1.2	Documentation	81
12.2	AdBlock	91
12.2.1	Problematic	91
12.2.2	Documentation	91
12.3	Auto-continue	92
12.3.1	Problematic	92
12.3.2	Documentation	92
12.4	Auto-save	92
12.4.1	Travis CI problematic	92
12.4.2	Documentation	93
12.5	Check	93
12.5.1	Problematic	93
12.5.2	Documentation	93
12.6	Cleaning	94
12.6.1	Problematic	94
12.6.2	Documentation	94
12.7	Configuration	94
12.7.1	Problematics	94
12.7.2	Documentation	95
12.8	Core	96
12.8.1	Documentation	96
12.9	Database	99
12.9.1	Problematics	99
12.9.2	Documentation	99
12.10	Directory Structure	101
12.10.1	Problematic	101
12.10.2	Documentation	101
12.11	Execution Time	101
12.11.1	Problematic	101
12.11.2	Documentation	102
12.12	Expiration Date	103
12.12.1	Problematic	103

12.12.2	Documentation	103
12.13	Generation	104
12.13.1	Problematic	104
12.13.2	Documentation	104
12.14	HTTP Code	105
12.14.1	Problematic	105
12.14.2	Documentation	105
12.15	IANA	105
12.15.1	Problematic	105
12.15.2	Documentation	105
12.16	Logs	106
12.16.1	Problematic	106
12.16.2	Documentation	106
12.17	Lookup	106
12.17.1	Problematics	106
12.17.2	Documentation	107
12.18	Mining	107
12.18.1	Problematic	107
12.18.2	Documentation	107
12.19	Percentage	108
12.19.1	Problematic	108
12.19.2	Documentation	108
12.20	Prints	108
12.20.1	Problematic	108
12.20.2	Documentation	108
12.21	Production	110
12.21.1	Problematic	110
12.21.2	Documentation	110
12.22	Public Suffix	110
12.22.1	Problematic	110
12.22.2	Documentation	110
12.23	Referer	111
12.23.1	Problematic	111
12.23.2	Documentation	111
12.24	Sort	111
12.24.1	Problematic	111
12.24.2	Documentation	111
12.25	Status	112
12.25.1	Problematic	112
12.25.2	Documentation	112
12.26	Syntax Checking	114
12.26.1	Problematic	114
12.26.2	Documentation	114
12.27	URL Testing	114
12.27.1	Problematic	114
12.27.2	Documentation	114
13	Known issues	115
14	Frequently Asked Questions	117
14.1	How to speed up a bit the process/test?	117
15	Contributing	119
15.1	Submitting changes	119

15.1.1	Steps before commit	119
15.1.2	The commit	120
15.2	Coding conventions	120
16	Contributors	121
17	Special Thanks	123
18	Supporting the project	125
19	They use PyFuncible!	127
20	License	129
21	Code of Conduct	131
21.1	Our Pledge	131
21.2	Our Standards	131
21.3	Our Responsibilities	132
21.4	Scope	132
21.5	Enforcement	132
21.6	Attribution	132
22	Discovered any issue?	133
23	Need help?	135
24	Indices and tables	137
	Python Module Index	139



PyFuncible is the little sister of [Funcible](#) which was archived on 13th March 2018.

Its main objective is to provide the availability of domains, IPs and since recently URL by generating an accurate result based on results from WHOIS, NSLOOKUP and HTTP status codes.

PyFuncible is currently running actively and daily with the help of Travis CI under 60+ repositories. It is used to clean or test the availability of data which are present in hosts files, list of IP, list of domains, block lists or even Adblock filter lists.

PyFuncible provides some useful features for continuous testing.

As an example, its auto-continue system coupled with its auto-save system allows it to run nice and smoothly under Travis CI without even reaching Travis CI time restriction. In the other side, its internal inactive database system let `INACTIVE` and `INVALID` caught domains, IPs or URLs being automatically retested over time on next run.

What can PyFunceble do?

- Test of the availability of a domain.
- Test of the availability of an IPv4.
- Test of the availability of an URL.
- Test of the availability of a domain/DNS name in a private or local network.
- Test of the availability of an IPv4 in a private or local network.
- Test of the availability of an URL in a private or local network.
- Test of the syntax of a domain.
- Test of the syntax of an IPv4.
- Test of the syntax of an URL.
- Test of domain or IP which are present into an Adblock formatted file.
- Test from a given raw link.
- Save test result(s) on file(s) (hosts file, plain text and/or JSON format).
- Show test result(s) on screen.
- Show percentage of each status (ACTIVE, INACTIVE, INVALID)
- Sort outputs hierarchically.
- “Mining” of domain or IP which are related to the tested element.
- Auto-continuation of tests in case of system crash or script stop.
- Filtering of a file content.
 - This feature will let us for example test all blogspot domain of the given file no matter the content of the file.
- Set the user-agent to use for the tests.

- Give some analytic depending of the HTTP status code (ACTIVE, POTENTIALLY_ACTIVE, POTENTIALLY_INACTIVE, SUSPICIOUS).
- Retest overtime of INACTIVE and INVALID domains.
- Print the execution time on screen and file.
- Customization of the different option via command-line arguments or configuration file.
- Continuous tests under Travis CI with the help of an auto saving and database system.
 - Set branch to push the result to for the autosaving system.
 - Set the minimal time before we autosave in order to avoid Travis CI limitation.
 - Set a command to execute at the end of the test.
 - Set the commit message for the autosaving system.
- ... and a lot more!

2.1 Domain testing

2.2 URL testing

2.3 Domain and URL testing with simple output

2.4 File content testing

2.5 File content testing with IDNA/Punycode conversion

2.6 Generated files/outputs

Dead-Hosts as place to use PyFunceble!

3.1 Why Dead-Hosts?

We are conscient that PyFunceble may run for days, that's why we offer you to request your list to be tested at [Dead-Hosts](#).

3.2 How do Dead-Hosts work?

[Dead-Hosts](#) is like a wrapper of PyFunceble. Indeed, we use Travis CI containers in order to test lists with PyFunceble.

Once a list is set up inside our infrastructure, PyFunceble will test the list regularly and the Dead-Hosts infrastructure will produce a `clean.list` file which represents the list of domains/IP/URL which remains or became `ACTIVE`.

In addition, if needed, we can set up a list of GitHub username to `@ping` once a new test is finished.

3.3 History of Dead-Hosts

The project started on 3rd March 2017 at [funilrys/dead-hosts](#).

The original idea was to test [Funceble](#) against hosts file in order to find bugs inside [Funceble](#) but also letting me have a fewer long hosts file.

On 23rd January 2018, I ([funilrys](#)) decided to shut [funilrys/dead-hosts](#) down. Indeed, as it became impossible to test all members of the project without having to wait weeks, I decided that it was time to move to another level.

That was the beginning of [Dead-Hosts](#).

Today [Dead-Hosts](#)'s objective is to provide to project/list maintainers or individuals - with the help of PyFunceble - more information about their favorite project/list or domains, IP or URL.

4.1 Requirements

Here is the list of requirements:

- Python 3.6+
- colorama
- domain2idna
- PyYAML
- requests
- setuptools
- urllib3

4.1.1 Python 3.6+

As we use for example

```
print('hello', end=' ')
```

which does not exist in Python 2.x and as I wanted to give a priority to Python 3, Python 3 is required.

4.1.2 colorama

As we use some coloration coloration, colorama is required.

4.1.3 domain2idna

As we propose the conversion of domains to IDNA, `domain2idna` is required.

Note: `domain2idna` is maintained and developed by [Nissar Chababy \(@funilrys\)](#), the main developer of PyFunceble. Its source code can be found [on GitHub](#).

4.1.4 PyYAML

As our configuration file is written in `.yaml`, `PyYAML` is required.

4.1.5 requests

As we use `requests` when calling all `Lookup()` methods, `requests` is required.

4.1.6 setuptools

As we use `install_requires=xx` inside our `setup.py`, `setuptools` is required.

4.1.7 urllib3

You should normally already have it. But as we handle some of its errors while using `requests`, `urllib3` is required.

4.2 Get PyFunceble

4.2.1 Stable version

Using pip

Choose your repository, install and enjoy PyFunceble!

From PyPi

```
$ pip3 install PyFunceble
```

From GitHub

```
$ pip3 install git+https://github.com/funilrys/PyFunceble.git@master#egg=PyFunceble
```

Using the AUR (for Arch Linux users)

The package can be found at <https://aur.archlinux.org/packages/pyfunceble/>.

With makepkg

```
$ wget https://aur.archlinux.org/cgit/aur.git/plain/PKGBUILD?h=pyfuncible
$ makepkg
$ sudo pacman -U pyfuncible*.tar.xz
```

With your favorite AUR helper

Warning: We do not recommend any AUR helper but keep in mind that some AUR helpers are “better” than other. For more information about your current (or any other) AUR helper please report to [the ArchWiki page](#).

```
$ yourFavoriteAurHelper -S pyfuncible
```

Pure Python method

Execute the following and enjoy PyFuncible!

```
$ git clone https://github.com/funilrys/PyFuncible.git
$ cd PyFuncible
$ python3 setup.py test && python3 setup.py install
```

4.2.2 Development version

The development version of PyFuncible represents the `dev` branch. It’s intended for the development of next features but is always at a usable state.

Indeed, We should not push to the `dev` branch until we are sure that the new commit does not break or introduce critical issue under PyFuncible.

For development

Execute the following and let’s hack PyFuncible!

Note: We highly recommend you to develop PyFuncible under a `virtualenv`.

```
$ git clone https://github.com/funilrys/PyFuncible.git
$ cd PyFuncible && git checkout dev && virtualenv venv
$ source venv/bin/activate && pip3 install -e .
```

Note: After installing with:

```
$ source venv/bin/activate && pip3 install -e .
```

- you only need to update the repository.
 - you don’t have to rerun the `pip` command.
-

For usage

Using pip

Execute one of the following and enjoy PyFuncible!

From PyPi

```
$ pip3 install PyFuncible-dev
```

From GitHub

```
$ pip3 install git+https://github.com/funilrys/PyFuncible.git@dev#egg=PyFuncible
```

Using the AUR (for Arch Linux users)

The package can be found at <https://aur.archlinux.org/packages/pyfuncible-dev/>.

With makepkg

```
$ wget https://aur.archlinux.org/cgit/aur.git/plain/PKGBUILD?h=pyfuncible-dev
$ makepkg
$ sudo pacman -U pyfuncible-dev*.tar.xz
```

With your favorite AUR helper

Warning: We do not recommend any AUR helper but keep in mind that some AUR helpers are “better” than other. For more information about your current (or any other) AUR helper please report to [the ArchWiki page](#).

```
$ yourFavoriteAurHelper -S pyfuncible-dev
```

Pure Python method

Execute the following and enjoy PyFuncible!

```
$ git clone https://github.com/funilrys/PyFuncible.git
$ cd PyFuncible && git checkout dev
$ python3 setup.py test && python3 setup.py install
```

4.3 First steps

Make sure that you can run

```
$ PyFuncible --version
```

and enjoy PyFuncible!!

5.1 Stable version

5.1.1 Using pip

Choose your repository, install and enjoy PyFuncible!

From PyPi

```
$ pip3 install --upgrade PyFuncible
```

From GitHub

```
$ pip3 install --upgrade git+https://github.com/funilrys/PyFuncible.git@master  
↪#egg=PyFuncible
```

5.1.2 Using the AUR (for Arch Linux users)

With makepkg

```
$ wget https://aur.archlinux.org/cgit/aur.git/plain/PKGBUILD?h=python-pyfuncible  
$ makepkg  
$ sudo pacman -U python-pyfuncible*.tar.xz
```

With your favorite AUR helper

Warning: We do not recommend any AUR helper but keep in mind that some AUR helpers are “better” than other. For more information about your current (or any other) AUR helper please report to [the ArchWiki page](#).

```
$ yourFavoriteAurHelper -Syu python-pyfuncible
```

5.1.3 Pure Python method

Execute the following and enjoy PyFuncible!

```
$ cd PyFuncible
$ git checkout master && git fetch origin && git merge origin/master
$ python3 setup.py test && python3 setup.py install
```

5.2 Development version

5.2.1 For development

```
$ cd PyFuncible && git checkout dev
$ git fetch origin && git merge origin/dev
```

Note: As you previously installed with

```
$ . venv/bin/activate && pip3 install -e .
```

Only code/repository update is required.

5.2.2 For usage

Using pip

Execute one of the following and enjoy PyFuncible!

From PyPi

```
$ pip3 install --upgrade PyFuncible-dev
```

From GitHub

```
$ pip3 install --upgrade git+https://github.com/funilrys/PyFuncible.git@dev
↪#egg=PyFuncible
```

Using the AUR (for Arch Linux users)

With makepkg

```
$ wget https://aur.archlinux.org/cgit/aur.git/plain/PKGBUILD?h=python-pyfuncible-dev
$ makepkg
$ sudo pacman -U python-pyfuncible-dev*.tar.xz
```

With your favorite AUR helper

Warning: We do not recommend any AUR helper but keep in mind that some AUR helpers are “better” than other. For more information about your current (or any other) AUR helper please report to [the ArchWiki page](#).

```
$ yourFavoriteAurHelper -Syu python-pyfuncible-dev
```

Pure Python method

Execute the following and enjoy PyFuncible!

```
$ cd PyFuncible && git checkout dev
$ git fetch origin && git merge origin/dev
$ python3 setup.py test && python3 setup.py install
```


This page will try to detail each configuration available into `.PyFunceble.yaml` along with the location of where we are looking for the configuration file.

6.1 Location

6.1.1 Problematics

- How can we create a more efficient way to work with configuration?
- How can we make the configuration file(s) available globally so that PyFunceble can be run everywhere in the user workspace?

To answer those problematics, we moved the configuration location elsewhere in the place where most users expect to have their configuration file(s).

6.1.2 Clone

If you cloned the repository and you're trying to test from a cloned directory (the one with for example `CONTRIBUTING.md`) we consider the configuration directory as the current one.

Note: This behavior allows us to not modify the way we develop PyFunceble.

6.1.3 Travis CI

Under [Travis CI](#), we search or initiate the configuration at the directory we are currently located.

Warning: We don't care about the distribution, as long as the `TRAVIS_BUILD_DIR` environment variable is set, we search or initiate the configuration in the current directory.

Note: If you want to force the directory where we should work, you can initiate the `PYFUNCEBLE_OUTPUT_DIR` environment variable with the path where we should work.

6.1.4 Linux and MacOS (Darwin Kernel)

Under Linux and MacOS, we look for the following directories in their order. If any configuration directory is found, the system proposes you to install them automatically on the first configuration file.

1. `~/ .config/PyFunceble`
2. `~/ .PyFunceble`
3. `${PWD}`

Note: If the parent directory does not exist, we move to the next possible location in the given order.

This means that under most Linux distributions and MacOS versions, we consider `~/ .config/PyFunceble` as the configuration location. But if the `~/ .config` directory does not exist, we fallback to `~/ .PyFunceble` as the configuration location.

6.1.5 Windows

As mentioned by [Pat Altimore's Blog](#), we used the Per user configuration files synchronized across domain joined machines via Active Directory Roaming section in order to understand what we should do to find our configuration directory.

Under Windows, we look for the following directories in their order. If any configuration directory is found, the system proposes you to install them automatically on the first configuration file.

1. `%APPDATA%\PyFunceble` (environnement variable)
2. `%CD%`

Note: `%CD%` is explained by the set command (`set /?`):

`%CD%` - expands to the current directory string.

Note: If the parent directory does not exist, we move to the next possible location in the given order.

This means that under most Windows versions, we consider `%APPDATA%\PyFunceble` - also know as `C:\Users\userName\AppData\Roaming\PyFunceble`- as the configuration location. But if the `%APPDATA%` directory does not exist, we fall back to the current directory as the configuration location.

6.1.6 Custom location

Sometimes, you may find yourself in a position where you absolutely do not want PyFuncible to use its default configuration location.

For that reason, if you set your desired configuration location along with the `PYFUNCIBLE_OUTPUT_DIR` environment variable, we take that location as the (default) configuration location.

6.2 Autoconfiguration

Sometimes, you may find yourself in a position that you do not or you can't answer the question which asks you if you would like to install the default configuration file.

For that reason, if you set `PYFUNCIBLE_AUTO_CONFIGURATION` as an environment variable with what you want an assignment, we do not ask that question. We simply do what we have to do without asking anything.

6.3 adblock

Type: `boolean`

Default value: `False`

Description: Enable / disable the adblock format decoding.

Note: If this index is set to `True`, every time we read a given file, we try to extract the elements that are present. We basically only decode the adblock format.

Note: If this index is set to `False`, every time we read a given file, we will consider one line as an element to test.

6.4 auto_continue

Type: `boolean`

Default value: `True`

Description: Enable / disable the auto continue system.

6.5 command

Type: `string`

Default value: `" "`

Description: Set the command to run before each commit (except the final one).

Note: The parsed command is called only if `auto_continue` and `travis` are set to `True`.

6.6 `command_before_end`

Type: `string`

Default value: `" "`

Description: Set the command to run before the final commit.

Note: The parsed command is called only if `auto_continue` and `travis` are set to `True`.

Note: Understand by final commit the commit which will deliver the last element we have to test.

6.7 `custom_ip`

Type: `string`

Default value: `"0.0.0.0"`

Description: Set the custom IP to use when we generate a line in the hosts file format.

Note: This index has no effect if `generate_hosts` is set to `False`.

6.8 `days_between_db_retest`

Type: `integer`

Default value: `1`

Description: Set the number of day(s) between each retest of the `INACTIVE` and `INVALID` elements which are present into `inactive_db.json`.

Note: This index has no effect if `inactive_database` is set to `False`.

6.9 `debug`

Type: `boolean`

Default value: `False`

Description: Enable / disable the generation of debug file(s).

Note: This index has no effect if `logs` is set to `False`

Warning: Do not touch this index unless you a have good reason to.

Warning: Do not touch this index unless you have been invited to.

6.10 filter

Type: string

Default value: ""

Description: Set the element to filter.

Note: This index should be initiated with a regular expression.

6.11 generate_hosts

Type: boolean

Default value: True

Description: Enable / disable the generation of the hosts file(s).

6.12 generate_json

Type: boolean

Default value: False

Description: Enable / disable the generation of the JSON file(s).

6.13 header_printed

Type: boolean

Default value: False

Description: Say to the system if the header has been already printed or not.

Warning: Do not touch this index unless you have a good reason to.

6.14 hierarchical_sorting

Type: boolean

Default value: False

Description: Say to the system if we have to sort the list and the outputs in a hierarchical order.

6.15 iana_whois_server

Type: string

Default value: whois.iana.org

Description: Set the server to call to get the whois referer of a given element.

Note: This index is only used when generating the `iana-domains-db.json` file.

Warning: Do not touch this index unless you have good reason to.

6.16 idna_conversion

Type: boolean

Default value: False

Description: Tell the system to convert all domains to IDNA before testing.

Note: We use `domain2idna` for the conversion.

6.17 inactive_database

Type: boolean

Default value: True

Description: Enable / Disable the usage of a database to store the INACTIVE and INVALID element to retest overtime.

6.18 less

Type: boolean

Default value: True

Description: Enable / Disable the output of every information of screen.

6.19 local

Type: boolean

Default value: False

Description: Enable / Disable the execution of the test(s) in a local or private network.

6.20 logs

Type: boolean

Default value: True

Description: Enable / Disable the output of all logs.

6.21 mining

Type: boolean

Default value: True

Description: Enable / Disable the mining subsystem.

6.22 no_files

Type: boolean

Default value: False

Description: Enable / Disable the generation of any file(s).

6.23 no_special

Type: boolean

Default value: False

Description: Enable / Disable the usage of the SPECIAL rules - which are describes in the source column section.

6.24 no_whois

Type: boolean

Default value: False

Description: Enable / Disable the usage of whois in the tests.

6.25 plain_list_domain

Type: `boolean`

Default value: `False`

Description: Enable / Disable the generation of the plain list of elements sorted by statuses.

Warning: Do not touch this index unless you have good reason to.

6.26 quiet

Type: `boolean`

Default value: `False`

Description: Enable / Disable the generation of output on the screen.

6.27 referer

Type: `string`

Default value: `" "`

Description: Set the referer of the element that is currently under test.

Warning: Do not touch this index unless you have good reason to.

6.28 seconds_before_http_timeout

Type: `integer`

Default value: `3`

Description: Set the timeout to apply to every HTTP status code request.

Note: This index must be a multiple of 3.

6.29 share_logs

Type: `boolean`

Default value: `True`

Description: Enable / disable the logs sharing.

Note: This index has no effect if `logs` is set to `False`.

6.30 `show_execution_time`

Type: `boolean`

Default value: `False`

Description: Enable / disable the output of the execution time.

6.31 `show_percentage`

Type: `boolean`

Default value: `True`

Description: Enable / disable the output of the percentage of each status.

6.32 `simple`

Type: `boolean`

Default value: `False`

Description: Enable / disable the simple output mode.

Note: If this index is set to `True`, the system will only return the result in format: `tested.element STATUS`.

6.33 `split`

Type: `boolean`

Default value: `True`

Description: Enable / disable the split of the results files.

Note: Understand with “results files” the mirror of what is shown on screen.

6.34 `syntax`

Type: `boolean`

Default value: `False`

Description: Enable / disable the syntax (only) testing.

Warning: If this index is set to `True`, we **ONLY** check for syntax, not availability.

6.35 `travis`

Type: `boolean`

Default value: `False`

Description: Enable / disable the Travis CI autosaving system.

Warning: Do not activate this index unless you are using PyFuncible under Travis CI.

6.36 `travis_autosave_commit`

Type: `string`

Default value: `"PyFuncible - AutoSave"`

Description: Set the default commit message we want to use when have to commit (save) but our tests are not yet completed.

6.37 `travis_autosave_final_commit`

Type: `string`

Default value: `"PyFuncible - Results"`

Description: Set the default final commit message we want to use when we all tests are finished.

6.38 `travis_autosave_minutes`

Type: `integer`

Default value: `15`

Description: Set the minimum of minutes we have to run before to automatically save our test results.

Note: As many services are setting a rate limit per IP, it's a good idea to set this value between 1 and 15 minutes.

6.39 `travis_branch`

Type: `string`

Default value: `master`

Description: Set the git branch where we are going to push our results.

6.40 unified

Type: boolean

Default value: False

Description: Enable / Disable the generation of the unified results.

Note: This index has no effect if `split` is set to `True`.

6.41 user_agent

Type: string

Default value: "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36"

Description: Set the User-Agent to use every time we are requesting something from a web server other than our API.

6.42 verify_ssl_certificate

Type: boolean

Default value: False

Description: Enable / Disable the verification of the SSL/TLS certificate when testing for URL.

Warning: If you set this index to `True`, you may get **false positive** result.

Indeed if the certificate is not registered to the CA or is simply invalid and the domain is still alive, you will always get `INACTIVE` as output.

6.43 whois_database

Type: boolean

Default value: True

Description: Enable / Disable the usage of the whois database to avoid/bypass whois server requests rate limit.

6.44 outputs

Type: dict

Description: Set the needed output tree/names.

Warning: If you choose to change anything please consider deleting our `output/` directory and the `dir_structure*.json` files.

6.44.1 `outputs[default_files]`

Type: dict

Description: Set the default name of some important files.

`outputs[default_files][dir_structure]`

Type: string

Default value: `dir_structure.json`

Description: Set the default filename of the file which has the structure to re-construct.

Note: This index has no influence with `dir_structure_production.json`

`outputs[default_files][iana]`

Type: string

Default value: `iana-domains-db.json`

Description: Set the default filename of the file which has the formatted copy of the IANA root zone database.

`outputs[default_files][inactive_db]`

Type: string

Default value: `inactive_db.json`

Description: Set the default filename of the file which will save the list of elements to retest overtime.

`outputs[default_files][results]`

Type: string

Default value: `results.txt`

Description: Set the default filename of the file which will save the formatted copy of the public suffix database.

`outputs[default_files][public_suffix]`

Type: string

Default value: `public-suffix.json`

Description: Set the default filename of the file which will save the mirror of what is shown on screen.

6.44.2 `outputs[domains]`

Type: dict

Description: Set the default name of some important files related to the `plain_list_domain` index.

`outputs[domains][directory]`

Type: string

Default value: `domains/`

Description: Set the default directory where we have to save the plain list of elements for each status.

`outputs[domains][filename]`

Type: string

Default value: `list`

Description: Set the default filename of the file which will save the plain list of elements.

6.44.3 `outputs[hosts]`

Type: dict

Description: Set the default name of some important files related to the `generate_hosts` index.

`outputs[hosts][directory]`

Type: string

Default value: `hosts/`

Description: Set the default directory where we have to save the hosts files of the elements for each status.

`outputs[hosts][filename]`

Type: string

Default value: `hosts`

Description: Set the default filename of the file which will save the hosts files of the elements.

6.44.4 `outputs[json]`

Type: dict

Description: Set the default name of some important files related to the `generate_json` index.

`outputs [json] [directory]`

Type: `string`

Default value: `json/`

Description: Set the default directory where we have to save the JSON files of the elements for each status.

`outputs [json] [filename]`

Type: `string`

Default value: `dump.json`

Description: Set the default filename of the file which will save the JSON files of the elements.

6.44.5 `outputs [analytic]`

Type: `dict`

Description: Set the default name of some important files and directories related to the `generate_hosts` index.

`outputs [analytic] [directories]`

Type: `dict`

Description: Set the default name of some important directories related to the `http_codes [active]` index.

`outputs [analytic] [directories] [parent]`

Type: `string`

Default value: `Analytic/`

Description: Set the default directory where we are going to put everything related to the HTTP analytic.

`outputs [analytic] [directories] [potentially_down]`

Type: `string`

Default value: `POTENTIALLY_INACTIVE/`

Description: Set the default directory where we are going to put all potentially inactive data.

`outputs [analytic] [directories] [potentially_up]`

Type: `string`

Default value: `POTENTIALLY_INACTIVE/`

Description: Set the default directory where we are going to put all potentially active data.

outputs [analytic] [directories] [up]

Type: string

Default value: POTENTIALLY_INACTIVE/

Description: Set the default directory where we are going to put all active data.

outputs [analytic] [directories] [suspicious]

Type: string

Default value: SUSPICIOUS/

Description: Set the default directory where we are going to put all suspicious data.

outputs [analytic] [filenames]

Type: dict

Description: Set the default name of some important files related to the `http_codes [active]` index and the HTTP analytic subsystem.

outputs [analytic] [filenames] [potentially_down]

Type: string

Default value: down_or_potentially_down

Description: Set the default filename where we are going to put all potentially inactive data.

outputs [analytic] [filenames] [potentially_up]

Type: string

Default value: potentially_up

Description: Set the default filename where we are going to put all potentially active data.

outputs [analytic] [filenames] [up]

Type: string

Default value: active_and_merged_in_results

Description: Set the default filename where we are going to put all active data.

outputs [analytic] [filenames] [suspicious]

Type: string

Default value: suspicious_and_merged_in_results

Description: Set the default filename where we are going to put all suspicious data.

6.44.6 `outputs[logs]`

Type: dict

Description: Set the default name of some important files and directories related to the `logs` index.

`outputs[logs][directories]`

Type: dict

Description: Set the default name of some important directories related to the `logs` index.

`outputs[logs][directories][date_format]`

Type: string

Default value: `date_format/`

Description: Set the default directory where we are going to put everything related to the data when the dates are in the wrong format.

`outputs[logs][directories][no_referer]`

Type: string

Default value: `no_referer/`

Description: Set the default directory where we are going to put everything related to the data when no referer is found.

`outputs[logs][directories][parent]`

Type: string

Default value: `no_referer/`

Description: Set the default directory where we are going to put everything related to the data when no referer is found.

`outputs[logs][directories][percentage]`

Type: string

Default value: `percentage/`

Description: Set the default directory where we are going to put everything related to percentages.

`outputs[logs][directories][whois]`

Type: string

Default value: `whois/`

Description: Set the default directory where we are going to put everything related to whois data.

Note: This is the location of all files when the `debug` index is set to `True`.

`outputs[logs][filenames]`

Type: dict

Description: Set the default filenames of some important files related to the `logs` index.

`outputs[logs][filenames][auto_continue]`

Type: string

Default value: `continue.json`

Description: Set the default filename where we are going to put the data related to the `auto continue` subsystem.

Note: This file is allocated if the `auto_continue` is set to `True`.

`outputs[logs][filenames][execution_time]`

Type: string

Default value: `execution.log`

Description: Set the default filename where we are going to put the data related to the execution time.

Note: This file is allocated if the `show_execution_time` is set to `True`.

`outputs[logs][filenames][percentage]`

Type: string

Default value: `percentage.txt`

Description: Set the default filename where we are going to put the data related to the percentage.

Note: This file is allocated if the `show_percentage` is set to `True`.

6.44.7 `outputs[main]`

Type: string

Default value: ""

Description: Set the default location where we have to generate the `parent_directory` directory and its dependencies.

6.44.8 `outputs[parent_directory]`

Type: `string`

Default value: `output/`

Description: Set the directory name of the parent directory which will contain all previously nounced directories.

6.44.9 `outputs[splited]`

Type: `dict`

Description: Set the default name of some important files and directory related to the `split` index.

`outputs[splited][directory]`

Type: `string`

Default value: `splited/`

Description: Set the default directory name where we are going to put the split data.

6.45 `status`

Type: `dict`

Description: Set the needed, accepted and status name.

6.45.1 `status[list]`

Type: `dict`

Description: Set the needed and accepted status name.

<p>Warning: All status should be in lowercase.</p>

`status[list][valid]`

Type: `list`

Default value: `["valid", "syntax_valid", "valid_syntax"]`

Description: Set the accepted `VALID` status.

Note: This status is only shown if the `syntax` index is activated.

status[list][up]

Type: list

Default value: ["up", "active"]

Description: Set the accepted ACTIVE status.

status[list][generic]

Type: list

Default value: ["generic"]

Description: Set the accepted generic status.

Note: This status is the one used to say the system that we have to print the complete information on the screen.

status[list][http_active]

Type: list

Default value: ["http_active"]

Description: Set the accepted status for the outputs[analytic][filenames][up] index.

status[list][down]

Type: list

Default value: ["down", "inactive", "error"]

Description: Set the accepted status INACTIVE index.

status[list][invalid]

Type: list

Default value: ["ouch", "invalid"]

Description: Set the accepted status INVALID index.

status[list][potentially_down]

Type: list

Default value: ["potentially_down", "potentially_inactive"]

Description: Set the accepted status for the outputs[analytic][filenames][potentially_down] index.

`status[list][potentially_up]`

Type: list

Default value: ["potentially_up", "potentially_active"]

Description: Set the accepted status for the outputs[analytic][filenames][potentially_up] index.

`status[list][suspicious]`

Type: list

Default value: ["strange", "hum", "suspicious"]

Description: Set the accepted status for the outputs[analytic][filenames][suspicious] index.

6.45.2 `status[official]`

Type: dict

Description: Set the official status name.

Note: Those status are the ones that are printed on the screen.

Warning: After any changes here please delete `dir_structure.json` and the `output/` directory.

`status[official][up]`

Type: string

Default value: ACTIVE

Description: Set the returned status for the ACTIVE case.

`status[official][down]`

Type: string

Default value: INACTIVE

Description: Set the returned status for the INACTIVE case.

`status[official][invalid]`

Type: string

Default value: INVALID

Description: Set the returned status for the INVALID case.

`status[official] [valid]`

Type: string

Default value: VALID

Description: Set the returned status for the VALID case.

Note: This status is only shown if the `syntax index` is activated.

6.46 http_codes

Type: dict

Description: Handle the interpretation of each status codes when we do and generate our analytic data.

6.46.1 http_codes[active]

Type: boolean

Default value: True

Description: Enable / Disable the usage of the HTTP status code extraction.

6.46.2 http_codes[list]

Type: dict

Description: Categorize the http status code as mentioned in the documentation related to the HTTP Code column.

`http_codes[list] [up]`

Type: list

Default value:

```
- 100
- 101
- 200
- 201
- 202
- 203
- 204
- 205
- 206
```

Description: List the HTTP status codes which are considered as ACTIVE.

`http_codes[list][potentially_down]`

Type: list

Default value:

```
- 400
- 402
- 403
- 404
- 409
- 410
- 412
- 414
- 415
- 416
```

Description: List the HTTP status code which are considered as INACTIVE or POTENTIALLY_INACTIVE.

`http_codes[list][potentially_up]`

Type: list

Default value:

```
- 000
- 300
- 301
- 302
- 303
- 304
- 305
- 307
- 403
- 405
- 406
- 407
- 408
- 411
- 413
- 417
- 500
- 501
- 502
- 503
- 504
- 505
```

Description: List the HTTP status code which are considered as ACTIVE or POTENTIALLY_ACTIVE.

6.47 links

Type: dict

Description: Set the list of links which can be used/called by the system when needed.

Note: The objective of this index is to avoid hardcoded links when the configuration file is readable.

6.47.1 `links[api_date_format]`

Type: string

Default value: `https://pyfuncible.funilrys.com/api/date-format`

Description: Set the link to use when we share logs.

6.47.2 `links[api_no_referer]`

Type: string

Default value: `https://pyfuncible.funilrys.com/api/no-referer`

Description: Set the link to use when we share logs.

6.47.3 `links[config]`

Type: string

Default value: `https://raw.githubusercontent.com/funilrys/PyFuncible/master/.PyFuncible_production.yaml`

Description: Set the upstream link to the configuration file.

6.47.4 `links[dir_structure]`

Type: string

Default value: `https://raw.githubusercontent.com/funilrys/PyFuncible/master/dir_structure_production.json`

Description: Set the upstream link to the directory structure dump file.

6.47.5 `links[iana]`

Type: string

Default value: `https://raw.githubusercontent.com/funilrys/PyFuncible/master/iana-domains-db.json`

Description: Set the upstream link to the IANA zone file configuration file.

6.47.6 `links[repo]`

Type: string

Default value: `https://github.com/funilrys/PyFuncible`

Description: Set the upstream link to the repository.

6.47.7 `links[requirements]`

Type: `string`

Default value: `https://raw.githubusercontent.com/funilrys/PyFuncible/master/requirements.txt`

Description: Set the upstream link to the `requirements.txt` file.

6.47.8 `links[ps1]`

Type: `string`

Default value: `https://raw.githubusercontent.com/funilrys/PyFuncible/master/public-suffix.json`

Description: Set the upstream link to the public suffix database file.

6.48 `counter`

Type: `dict`

Description: Setup the internal counter.

Warning: The following is not intended for modification. Exception for debugging or special cases which requires an initiated counter.

6.48.1 `counter[number]`

Type: `dict`

Description: Setup the internal counter.

Warning: The following is not intended for modification. Exception for debugging or special cases which requires an initiated counter.

`counter[number][down]`

Type: `integer`

Default value: `0`

Description: Setup the internal down counter.

Warning: The following is not intended for modification. Exception for debugging or special cases which requires an initiated counter.

`counter[number] [invalid]`

Type: `integer`

Default value: 0

Description: Setup the internal invalid counter.

Warning: The following is not intended for modification. Exception for debugging or special cases which requires an initiated counter.

`counter[number] [tested]`

Type: `integer`

Default value: 0

Description: Setup the internal tested counter.

Warning: The following is not intended for modification. Exception for debugging or special cases which requires an initiated counter.

`counter[number] [up]`

Type: `integer`

Default value: 0

Description: Setup the internal up counter.

Warning: The following is not intended for modification. Exception for debugging or special cases which requires an initiated counter.

6.48.2 `counter[percentage]`

Type: `dict`

Description: Setup the internal percentage counter.

Warning: The following is not intended for modification. Exception for debugging or special cases which requires an initiated counter.

`counter[percentage] [down]`

Type: `integer`

Default value: 0

Description: Setup the internal down percentage counter.

Warning: The following is not intended for modification. Exception for debugging or special cases which requires an initiated counter.

`counter [percentage] [invalid]`

Type: `integer`

Default value: 0

Description: Setup the internal invalid percentage counter.

Warning: The following is not intended for modification. Exception for debugging or special cases which requires an initiated counter.

`counter [percentage] [up]`

Type: `integer`

Default value: 0

Description: Setup the internal up percentage counter.

Warning: The following is not intended for modification. Exception for debugging or special cases which requires an initiated counter.

7.1 From a terminal

7.1.1 Detailed

Note: `False` stand for deactivated when `True` stand for activated.

`-ad | --adblock`

Switch the decoding of the adblock format.

Default value: `False`

If this argument is activated the system will extract all domains or IP from the given adblock file.

`-a | --all`

Output all available information on the screen.

Default value: `False`

When activated:

Domain	Status	Expiration Date	Source	HTTP Code
pyfunceble.readthedocs.io	ACTIVE	Unknown	NSLOOKUP	302

When deactivated:

Domain	Status	HTTP Code
-----	-----	-----
pyfuncible.readthedocs.io	ACTIVE	302

-c | --auto-continue | --continue

Switch the value of the auto continue mode.

Default value: True

This argument activates or deactivates the auto-continue subsystem. Indeed, as we can automatically continue if the script has been stopped, this switch allows us to disable or enable the usage of that specific subsystem.

--clean

Clean all files under output.

As it is sometimes needed to clean our `output/` directory, this argument does the job automatically.

Warning: This argument delete everything which are `.keep` or `.gitignore`

--clean-all

Clean all files under output and all file generated by PyFuncible.

Warning: This really deletes everything we generated without any warning.

--cmd "something"

Pass a command before each commit (except the final one).

Default value: ''

Note: In this example, `something` should be a script or a program which have to be executed when we reached the end of the given file.

Note: This argument is only used if `--travis` or `travis : true` (under `.PyFuncible.yaml`) are activated.

--cmd-before-end "something"

Pass a command before the results (final) commit under the Travis mode.

Default value: ''

Note: In this example, `something` should be a script or a program which have to be executed when we reached the end of the given file.

Note: This argument is only used if `--travis` or `travis : true` (under `.PyFunceble.yaml`) are activated.

`--commit-autosave-message "something"`

Replace the default autosave commit message.

Default value: `PyFunceble - AutoSave`

This argument allows us to set a custom commit message which is going to be used as commit message when saving.

Note: This argument is only used if `--travis` or `travis : true` (under `.PyFunceble.yaml`) are used.

Note: This argument is only used if we have to split the work into multiple processes because a list is too long or the timeout is reached.

Warning: Please avoid the usage of `[ci skip]` here.

`--commit-results-message "something"`

Replace the default results (final) commit message.

Default value: `PyFunceble - Results`

Note: This argument is only used if `--travis` or `travis : true` (under `.PyFunceble.yaml`) are used.

Note: This argument is only used if we reached the end of the list we are or have to test.

`-d "something" | --domain "something"`

Set and test the given domain.

This argument will test and give the results of the tests of the given domain.

Note: For this argument (and only for this argument), we are converting the given string to lowercase.

-db | --database

Switch the value of the usage of a database to store inactive domains of the currently tested list.

Default value: `True`

This argument will disable or enable the usage of a database which saves all *INACTIVE* and *INVALID* domain of the given file over time.

Note: The database is retested every x day(s), where x is the number set in `-dbr "something"`.

-dbr "something"

Set the numbers of days between each retest of domains present into the database of *INACTIVE* and *INVALID* domains.

Default value: `1`

Note: This argument is only used if `-db` or `inactive_database : true` (under `.PyFuncible.yaml`) are activated.

--debug

Switch the value of the debug mode.

Default value: `False`

This argument activates the debug mode. Under the debug mode, everything caught by the whois subsystem is saved.

Warning: Do not use this argument unless you have been told to.

--directory-structure

Generate the directory and files that are needed and which does not exist in the current directory.

Want to start without anything? This argument generates the output directory automatically for you!

Note: In case of a file or directory not found issue, it's recommended to remove the `dir_structure.json` along with the `output/` directory before using this argument.

-ex | --execution

Switch the default value of the execution time showing.

Default value: `False`

Want to know the execution time of your test? Well, this argument will let you know!

`-f "something" | --file "something"`

Read the given file and test all domains inside it. If a URL is given we download and test the content of the given URL.

Note: We consider one line as one domain or one commented line. A line can be commented at the end.

Note: You can give a raw link and the system will download and test its content.

`--filter "something"`

Domain to filter (regex).

Want to test all `blogspot` from your list? This argument allows you to do that!

Note: This argument should be a regex expression.

`--help`

Show the help message and exit.

`-h | --host`

Switch the value of the generation of hosts file.

Default value: `True`

This argument will let the system know if it has to generate the hosts file version of each status.

`--hierarchical`

Switch the value of the hierarchical sorting of the tested file.

Default value: `True`

This argument will let the system know if we have to sort the list and our output in hierarchical order.

`--http`

Switch the value of the usage of HTTP code.

Default value: `True`

You don't want to take the result of the HTTP code execution in consideration? This argument allows you to disable that!

Note: If activated the subsystem will bypass the HTTP status code extraction `logic-representation.rst`

--iana

Update/Generate *iana-domains-db.json*.

This argument generates or updates *iana-domains-db.json*.

--idna

Switch the value of the IDNA conversion.

Default value: False

This argument allows the conversion of the domains using `domain2idna`

Warning: This feature is not supported yet for the URL testing.

-ip "something"

Change the IP to print with the hosts files.

Default value: 0.0.0.0

--json

Switch the value of the generation of the JSON formatted list of domains.

Default value: False

--less

When activated:

Domain	Status	HTTP Code
pyfuncible.readthedocs.io	ACTIVE	302

When deactivated:

Domain	Status	Expiration Date	Source	HTTP Code
pyfuncible.readthedocs.io	ACTIVE	Unknown	NSLOOKUP	302

--local

Switch the value of the local network testing.

Default value: False

Want to run a test over a local or private network? This argument will disable the limitation which does not apply to private networks.

--link "something"

Download and test the given file.

Want to test a raw link? This argument will download and test the given raw link.

-m | --mining

Switch the value of the mining subsystem usage.

Default value: False

Want to find domain or URL linked to a domain in your list? This argument will exactly do that.

-n | --no-files

Switch the value the production of output files.

Default value: False

Want to disable the production of the outputted files? This argument is for you!

-nl | --no-logs

Switch the value of the production of logs files in the case we encounter some errors.

Default value: False

Don't want any logs to go out of PyFuncible? This argument disables every logs subsystem.

-ns | --no-special

Switch the value of the usage of the SPECIAL rules.

Default value: False

Don't want to use/apply the SPECIAL rules - which are explained in the source column section ? This argument disable them all.

-nu | --no-unified

Switch the value of the production unified logs under the output directory.

Default value: True

This argument disables the generation of *result.txt*.

`-nw | --no-whois`

Switch the value the usage of whois to test domain's status.

Default value: `False`

Don't want to use or take in consideration the results from `whois`? This argument allows you to disable it!

`-p | --percentage`

Switch the value of the percentage output mode.

Default value: `True`

This argument will disable or enable the generation of the percentage of each status.

`--plain`

Switch the value of the generation of the plain list of domains.

Default value: `False`

Want to get a list with all domain for each status? The activation of this argument does the work while testing!

`--production`

Prepare the repository for production.

<p>Warning: Do not use this argument unless you have been told to, you prepare a Pull Request or you want to distribute your modified version of PyFuncible.</p>

`-ps1 | --public-suffix`

Update/Generate *public-suffix.json*.

This argument will generate or update *public-suffix.json*.

`-q | --quiet`

Run the script in quiet mode.

Default value: `False`

You prefer to run a program silently? This argument is for you!

`--share-logs`

Switch the value of the sharing of logs.

Default value: `True`

Want to make PyFuncible a better tool? Share your logs with our API which collect all logs!

`-s | --simple`

Switch the value of the simple output mode.

Default value: `False`

Want as less as possible data on screen? This argument returns as less as possible on screen!

`--split`

Switch the value of the split of the generated output

Default value: `True`

Want to get the logs (copy of what you see on screen) on different files? This argument is suited to you!

`--syntax`

Switch the value of the syntax test mode.

Default value: `False`

`-t "something" | --timeout "something"`

Switch the value of the timeout.

Default value: `3`

This argument will set the default timeout to apply everywhere it is possible to set a timeout.

`--travis`

Switch the value of the Travis mode.

Default value: `False`

Want to use PyFuncible under Travis CI? This argument is suited for your need!

`-url "something" | --url "something"`

Analyze the given URL.

Want to test the availability or an URL? Enjoy this argument!

Note: When we test the availability of an URL, we check the HTTP status code of the given URL.

`-uf "something" | --url-file "something"`

Read and test the list of URL of the given file. If a URL is given we download and test the content of the given URL.

Note: We consider one line as one URL to test.

Note: You can give a raw link and the system will download and test its content.

-ua "something" | --user-agent "something"

Set the user-agent to use and set every time we interact with everything which is not our logs sharing system.

-v | --version

Show the version of PyFuncible and exit.

-vsc | --verify-ssl-certificate

Switch the value of the verification of the SSL/TLS certificate when testing for URL.

Default value: False

Warning: If you activate the verification of the SSL/TLS certificate, you may get **false positive** results.

Indeed if the certificate is not registered to the CA or is simply invalid and the domain is still alive, you will always get `INACTIVE` as output.

-wdb | --whois-database

Switch the value of the usage of a database to store whois data in order to avoid whois servers rate limit.

Default value: True

7.1.2 Global overview

```
usage: PyFuncible [-ad] [-a] [-c] [--autosave-minutes AUTOSAVE_MINUTES]
                 [--clean] [--clean-all] [--cmd CMD]
                 [--cmd-before-end CMD_BEFORE_END]
                 [--commit-autosave-message COMMIT_AUTOSAVE_MESSAGE]
                 [--commit-results-message COMMIT_RESULTS_MESSAGE]
                 [-d DOMAIN] [-db] [-dbr DAYS_BETWEEN_DB_RETEST] [--debug]
                 [--directory-structure] [-ex] [-f FILE] [--filter FILTER]
                 [--help] [--hierarchical] [-h] [--http] [--iana] [--idna]
                 [-ip IP] [--json] [--less] [--local] [--link LINK] [-m] [-n]
                 [-nl] [-ns] [-nu] [-nw] [-p] [--plain] [--production] [-psl]
                 [-q] [--share-logs] [-s] [--split] [--syntax] [-t TIMEOUT]
                 [--travis] [--travis-branch TRAVIS_BRANCH] [-u URL]
                 [-uf URL_FILE] [-ua USER_AGENT] [-v] [-vsc] [-wdb]
```

(continues on next page)

(continued from previous page)

```

optional arguments:
  -ad, --adblock          Switch the decoding of the adblock format.
                          Configured value: False
  -a, --all               Output all available information on the screen.
                          Configured value: True
  -c, --auto-continue, --continue
                          Switch the value of the auto continue mode.
                          Configured value: True
  --autosave-minutes AUTOSAVE_MINUTES
                          Update the minimum of minutes before we start
                          committing to upstream under Travis CI.
                          Configured value: 15
  --clean                 Clean all files under output.
  --clean-all            Clean all files under output and all file generated by
                          PyFuncible.
  --cmd CMD                Pass a command to run before each commit (except the
                          final one) under the Travis mode. Configured
                          value: ''
  --cmd-before-end CMD_BEFORE_END
                          Pass a command to run before the results (final)
                          commit under the Travis mode. Configured
                          value: ''
  --commit-autosave-message COMMIT_AUTOSAVE_MESSAGE
                          Replace the default autosave commit message.
                          Configured value: 'PyFuncible -
                          AutoSave'
  --commit-results-message COMMIT_RESULTS_MESSAGE
                          Replace the default results (final) commit message.
                          Configured value: 'PyFuncible -
                          Results'
  -d DOMAIN, --domain DOMAIN
                          Set and test the given domain.
  -db, --database          Switch the value of the usage of a database to store
                          inactive domains of the currently tested list.
                          Configured value: True
  -dbr DAYS_BETWEEN_DB_RETEST, --days-between-db-retest DAYS_BETWEEN_DB_RETEST
                          Set the numbers of days between each retest of domains
                          present into inactive-db.json. Configured
                          value: 1
  --debug                 Switch the value of the debug mode.
                          Configured value: False
  --directory-structure
                          Generate the directory and files that are needed and
                          which does not exist in the current directory.
  -ex, --execution        Switch the default value of the execution time
                          showing. Configured value: False
  -f FILE, --file FILE    Read the given file and test all domains inside it. If
                          a URL is given we download and test the content of the
                          given URL.
  --filter FILTER         Domain to filter (regex).
  --help                  Show this help message and exit.
  --hierarchical          Switch the value of the hierarchical sorting of the
                          tested file. Configured value: False
  -h, --host              Switch the value of the generation of hosts file.
                          Configured value: True
  --http                  Switch the value of the usage of HTTP code.
                          Configured value: True

```

(continues on next page)

(continued from previous page)

```

--iana                Update/Generate `iana-domains-db.json`.
--idna                Switch the value of the IDNA conversion.
                    Configured value: False
-ip IP                Change the IP to print in the hosts files with the
                    given one. Configured value:
                    '0.0.0.0'
--json                Switch the value of the generation of the JSON
                    formatted list of domains. Configured value:
                    False
--less                Output less informations on screen.
                    Configured value: False
--local                Switch the value of the local network testing.
                    Configured value: True
--link LINK           Download and test the given file.
-m, --mining           Switch the value of the mining subsystem usage.
                    Configured value: False
-n, --no-files        Switch the value of the production of output files.
                    Configured value: False
-nl, --no-logs        Switch the value of the production of logs files in
                    the case we encounter some errors. Configured
                    value: False
-ns, --no-special     Switch the value of the usage of the SPECIAL rules.
                    Configured value: False
-nu, --no-unified     Switch the value of the production unified logs under
                    the output directory. Configured value:
                    False
-nw, --no-whois       Switch the value the usage of whois to test domain's
                    status. Configured value: False
-p, --percentage      Switch the value of the percentage output mode.
                    Configured value: True
--plain                Switch the value of the generation of the plain list
                    of domains. Configured value: False
--production           Prepare the repository for production.
-psl, --public-suffix
                    Update/Generate `public-suffix.json`.
-q, --quiet            Run the script in quiet mode. Configured
                    value: False
--share-logs           Switch the value of the sharing of logs.
                    Configured value: True
-s, --simple            Switch the value of the simple output mode.
                    Configured value: False
--split                Switch the value of the split of the generated output
                    files. Configured value: True
--syntax                Switch the value of the syntax test mode.
                    Configured value: False
-t TIMEOUT, --timeout TIMEOUT
                    Switch the value of the timeout. Configured
                    value: 3
--travis                Switch the value of the Travis mode.
                    Configured value: False
--travis-branch TRAVIS_BRANCH
                    Switch the branch name where we are going to push.
                    Configured value: 'master'
-u URL, --url URL      Analyze the given URL.
-uf URL_FILE, --url-file URL_FILE
                    Read and test the list of URL of the given file. If a
                    URL is given we download and test the content of the

```

(continues on next page)

(continued from previous page)

```

        given URL.
-ua USER_AGENT, --user-agent USER_AGENT
        Set the user-agent to use and set every time we
        interact with everything which is not our logs sharing
        system.
-v, --version
        Show the version of PyFuncible and exit.
-vsc, --verify-ssl-certificate
        Switch the value of the verification of the SSL/TLS
        certificate when testing for URL. Configured
        value: False
-wdb, --whois-database
        Switch the value of the usage of a database to store
        whois data in order to avoid whois servers rate limit.
        Configured value: True

```

Crafted with `by` Nissar Chababy (Funilrys) with the help of <https://pyfuncible.rtfid.io/en/master/contributors.html> && <https://pyfuncible.rtfid.io/en/master/special-thanks.html>

7.2 From a Python script or module

Before continuing reading this part, You should know that I consider that you can speak in Python. If it's not the case, well, it's the time to [learn Python!](#)

As **PyFuncible** is written in Python, it can be easily imported and used inside a script or another module.

This section will present some examples.

7.2.1 Availability check of domains, IP or URL

```

"""
This is a basic example which prints one of the availability of
the given domain and URL.

.. note:
    Official output: ACTIVE, INACTIVE, INVALID
"""

from PyFuncible import test as PyFuncible
from PyFuncible import url_test as PyFuncibleURL

DOMAIN, IP = ("github.com", "103.86.96.100")
URL = "https://{}/".format(DOMAIN)

print(DOMAIN, PyFuncible(domain=DOMAIN))
print(URL, PyFuncibleURL(url=URL))
print(IP, PyFuncible(domain=IP))

```

7.2.2 Syntax check of domains, IP or URL

```

"""
This is a basic example which checks syntax of the given element.
"""

from PyFunceble import syntax_check as PyFuncebleDomainSyntax
from PyFunceble import url_syntax_check as PyFuncebleURLSyntax
from PyFunceble import ipv4_syntax_check as PyFuncebleIPv4Syntax

print("google.com", PyFuncebleDomainSyntax(domain="google.com"))
print("https://google.com", PyFuncebleURLSyntax(url="https://google.com"))
print("216.58.207.46", PyFuncebleIPv4Syntax(ip="216.58.207.46"))

print("forest-jump", PyFuncebleDomainSyntax(domain="forest-jump"))
print("https://forest-jump", PyFuncebleURLSyntax(url="https://forest-jump"))
print("257.58.207.46", PyFuncebleIPv4Syntax(ip="257.58.207.46"))

```

7.2.3 IPv4 Range and subdomain syntax check

```

"""
This is a basic example which checks syntax of the given element.
"""

from PyFunceble import is_ipv4_range, is_subdomain

print("hello.google.com", is_subdomain(domain="hello.google.com"))
print("google.com", is_subdomain(domain="google.com"))

print("192.168.0.0/24", is_ipv4_range(ip="192.168.0.0/24"))
print("192.168.0.0", is_ipv4_range(ip="192.168.0.0"))

```

7.2.4 Loop example

This part is unnecessary but we wanted to document it!!

```

"""
This is a loop example which tests a list of domain and processes some action
according to one of the official output of PyFunceble.

..note:
* Official output: ACTIVE, INACTIVE, INVALID
* You should always use PyFunceble().test() as it's the method which is especially
  suited for `__name__ != '__main__'` usage.
"""

from PyFunceble import test as PyFunceble
from PyFunceble import url_test as PyFuncebleURL

DOMAINS = ["twitter.com", "google.com", "github.com", "github.comcomcom", "funilrys.co
↵"]

def domain_status(domain_or_ip):
    """
    Check the status of the given domain name or IP.

```

(continues on next page)

(continued from previous page)

```

Argument:
    - domain_or_ip: str
      The domain or IPv4 to test.

Returns: str
    The status of the domain.
"""

return PyFuncible(domain_or_ip)

def url_status(url):
    """
    Check the status of the given url.

    Argument:
        - url: str
          The URL to test.

    Returns: str
        The status of the URL.
    """

    return PyFuncibleURL(url)

for domain in DOMAINS:
    print(
        "%s is %s and %s is %s"
        % (
            domain,
            domain_status(domain),
            "http://" + domain,
            url_status("http://" + domain),
        )
    )

```

7.2.5 Advanced example

PyFuncible allow you to get the following information as a dictionary. The objective behind this feature is to let you know more about the element you are testing.

```

{
    "_status": None, # If some extra rules are applied, this index will keep the
↳status before the extra rules was applied.
    "_status_source": None, # If some extra rules are applied, this index will keep
↳the source before the extra rules was applied.
    "domain_syntax_validation": None, # The domain syntax validation status.
    "expiration_date": None, # The expiration_date of the element if found.
    "http_status_code": None, # The status code of the tested element.
    "ip4_syntax_validation": None, # The IPv4 syntax validation status.
    "nslookup": [], # A list of IP of the tested element.
    "status_source": None, # The source which gives us the status.
    "status": None, # The status matched by PyFuncible.
}

```

(continues on next page)

(continued from previous page)

```

"tested": None, # The tested element.
"url_syntax_validation": None, # The url syntax validation status.
"whois_record": None, # The whois record if whois_server is found.
"whois_server": None, # The whois server we use to get the whois record (if_
→found).
}

```

To get that information simply work with our interface like follow :)

```

"""
This is an advanced example which get more information about the tested element.
"""

from PyFuncible import test as PyFuncible
from PyFuncible import url_test as PyFuncibleURL

DOMAIN = "google.com"

DOMAIN_RESULT_FROM_API = PyFuncible(domain=DOMAIN, complete=True)
URL_RESULT_FROM_API = PyFuncibleURL(url="https://{}/".format(DOMAIN), complete=True)

print("nslookup", DOMAIN_RESULT_FROM_API["nslookup"])
print("domain_syntax_validation", DOMAIN_RESULT_FROM_API["domain_syntax_validation"])
print(DOMAIN_RESULT_FROM_API["tested"], DOMAIN_RESULT_FROM_API["status"])

print("nslookup", URL_RESULT_FROM_API["nslookup"])
print("domain_syntax_validation", URL_RESULT_FROM_API["domain_syntax_validation"])
print("url_syntax_validation", URL_RESULT_FROM_API["url_syntax_validation"])
print(URL_RESULT_FROM_API["tested"], DOMAIN_RESULT_FROM_API["status"])

```

7.2.6 Custom Configuration

Sometime you may want to change **PyFuncible**'s configuration information from within your code. Here are way to do it.

```

"""
This is an example about how we can update the configuration while developping on top
of PyFuncible.
"""

import PyFuncible
from PyFuncible import test as PyFuncibleTest

# We preset the indexes (from .PyFuncible.yaml) that we want to update.
CUSTOM_CONFIGURATION_INDEX_VALUE_TO_SET = {"no_whois": True}

# We parse our custom indexes to PyFuncible before starting to use it.
PyFuncible.load_config(custom=CUSTOM_CONFIGURATION_INDEX_VALUE_TO_SET)

# From now, each call of test so in this example PyFuncibleTest,
# will not try to get/request the WHOIS record.

DOMAINS = ["google.com", "github.com"]

print("Start with global custom configuration.")

```

(continues on next page)

(continued from previous page)

```

for DOMAIN in DOMAINS:
    # This should return None
    print(DOMAIN, PyFuncibleTest(domain=DOMAIN, complete=True) ["whois_record"])
print("End with global custom configuration.\n")

print("Start with local custom configuration.")

# We update our index so that we can test/see how to parse it locally.
CUSTOM_CONFIGURATION_INDEX_VALUE_TO_SET["no_whois"] = False

for DOMAIN in DOMAINS:
    print("Start of WHOIS record of %s \n\n" % DOMAIN)

    # This part should return the WHOIS record.

    # This will - at each call of PyFuncibleTest or PyFuncibleURLTest on url testing -
    # update the configuration data with the one you give.
    print(
        PyFuncibleTest(
            domain=DOMAIN, complete=True, config=CUSTOM_CONFIGURATION_INDEX_VALUE_TO_
↪SET
        ) ["whois_record"]
    )
    print("\n\nEnd of WHOIS record of %s" % DOMAIN)
print("End with local custom configuration.")

```

7.3 From Travis CI

As we offer an argument named `--travis` to activate the usage of PyFuncible in a Travis CI instance, we document here what you need to know!

7.3.1 Configuration

Note: This part only present a commented `.travis.yml` so that you can understand where to start.

If you need more practical examples, feel free to report to one of [Dead-Hosts](#) repositories which use PyFuncible with Travis CI.

```

env:
  global:
    # The following is your encrypted GitHub API key.
    # Indeed as we are going to push to the repository, this is needed.
    - secure: QQdKFquFFo jFT9XJlXZp4EMoDTVVoXFgqZq8XU+sCVf+pJQR6d/
↪oKBp8rnSTCnZizWQQUXjGXUUXUpSG/
↪dYGYBLjo3rH3rsn9ciZHVfubxbwK860w4sqibl4DvhCv2rdsFtvzXnhm4P90L3i+krKdewh9fxpNyUU58qOgfnS7mK9FcFhb8z
↪dc0y36q6vjfgJR+QchetOtHgNbKYbLB8c26Di90OZCFJsxMNcl1Wct4qFPXkFGvjXrISW6pbdPL5Plto0Iq3iLiulhYOPVAryst
↪67d+F9ocQdIoFpDDTdgIjHerQnD2wRglaKPzLDb4jJTpqr5ssPrqUAKl3st7gyaAZzCEADPDnIBDjOJS+mFWbx9DKgc=
    # This is the Git name we have to set. (git config user.name)
    - GIT_NAME: Travis CI
    # This is the Git Email we have to set. (git config user.email)
    - GIT_EMAIL: dead-hosts@funilrys.com

```

(continues on next page)

(continued from previous page)

```
# This is the full slug of the repository we are working with.
- TRAVIS_REPO_SLUG: dead-hosts/repository-structure
# This is the branch we have to checkout and push to.
- GIT_BRANCH: master

# This is the language we use.
language: python

# This is the python version we are going to use for the tests.
# Note: you can add any 3.x version to the list.
python:
- "3.6"

# The following will tell Travis CI to ends as fast as possible.
matrix:
  fast_finish: true

# Here we are setting what Travis CI have to cache.
cache:
  # We are caching pip3 as we use it to install PyFuncible
  - pip3

install:
  # We install the development version of PyFuncible. If you prefer the stable_
↪version replace `pyfuncible-dev` with `pyfuncible`.
  - pip3 install pyfuncible-dev

# Our tests start here.
script:
  # Let's say we want our results and our PyFuncible infrastructure to be saved in_
↪a directory called `PyFuncible-tests`

  # We move inside it.
  - cd PyFuncible-tests
  # We test the file `my_awesome_list` which is located inside the current_
↪directory.
  # Note: we precise the `--travis` argument here,
  #       but you work without it if you set `travis: true` inside your `PyFuncible.
↪yaml`
  - PyFuncible --travis -f my_awesome_list --plain

# The following initiate email notification logic.
notifications:
  # As we want to get a mail on failure and on status change, we set the following.
  on_success:  change
  on_failure:  always
```

7.3.2 Getting a GitHub token

For the secure index of the `.travis.yml` file, you have to generate a new GitHub token.

After you got your token, please write it or save it in a safe place as you're going to need it every time you're going to interact with Travis CI.

Note: The scope to set is `public_repo` but you can also set others depending on your needs.

7.3.3 Encrypting the token for Travis CI usage

To encrypt the token simply replace and execute the following according to your personal case.

```
$ travis encrypt 'GH_TOKEN=theGeneratedToken' -r 'The content of TRAVIS_REPO_SLUG' --  
↪add
```

Warning: Please do not execute the following explicitly without replacing `theGeneratedToken` with your previously generated GitHub token and `The content of TRAVIS_REPO_SLUG` with your repository slug.

Note: The usage of `--add` ensure that the `travis` program automatically add the secure index to the `.travis.yml` file.

Because PyFuncible provides multiple information in a table, we chose to document them all. The objective of this page is to reflect what the code actually do in a more clear and understandable way.

8.1 Domains

This column is one of the basic ones as it gives us the name of the last tested domain or IP.

8.2 Status

There's 3 possible output for this column.

8.2.1 ACTIVE

This status is returned when **one of the following cases** is met:

- We can extract the expiration date from `Lookup().whois()`.
 - *Please note that we don't check if the date is in the past.*
- `Lookup().nslookup()` don't return an error.
 - *Please note that we don't read the returned value.*
- `HTTPCode().get()` return one the following code [100, 101, 200, 201, 202, 203, 204, 205, 206].

8.2.2 INACTIVE

This status is returned when **all the following cases** are met:

- We can't extract the expiration date from `Lookup().whois()`.
- `Lookup().nslookup()` don't return an error.

8.2.3 INVALID

This status is returned when **all the following cases** are met:

- Domain/IP does not match pass our syntax checker.
- Domain extension is unregistered in [IANA Root Zone Database](#).

Note: Understand by this that the extension is not present in the `iana-domains-db.json` file.

8.2.4 VALID

This status is returned when we are checking for syntax. It is the equivalent of `ACTIVE` but for syntax checking.

8.3 Expiration Date

There's two possible output for this column.

8.3.1 Unknown

Unknown is returned when we could not extract the expiration date from `Lookup().whois()` outputs.

8.3.2 A date

Only if we could extract the expiration date from `Lookup().whois()`, the date becomes formatted like `02-jan-2017`.

8.4 Source

At this time, there's 5 possible output for this column.

8.4.1 HTTP Code

This source is returned when **all the following cases** are met:

- We can't extract the expiration date from `Lookup().whois()`.
- The `INACTIVE` status is the one returned by other methods.
- `HTTPCode().get()` outputs is one the following `[100, 101, 200, 201, 202, 203, 204, 205, 206]`.

8.4.2 IANA

This source is always returned when the domain has the status `INVALID`. The usage of this source comes from the comparison of the element extension against the [IANA Root Zone Database](#).

8.4.3 SYNTAX

This source is always returned when the domain has the status `INVALID` or in the case that we are only checking for syntax instead of availability. The usage of this source comes from the comparison of the element against our domain, IP or URL syntax validation system.

8.4.4 NSLOOKUP

This source is always returned when the taken decision of the status of the domain/IP comes from `Lookup()`. `nslookup()` outputs.

8.4.5 SPECIAL

As `PyFuncible` grows, I thought that a bit of filtering for special cases would be great. So I introduced the `SPECIAL` source.

Note: Please consider all 3 digits number that are listed in this section as the HTTP status code caught by `HTTPCode().get()`.

Warning: Do not want those rules? You can use following to disable them.

- `-ns`|:code=no-special`` arguments from the CLI.
- `no_special: True` into your local configuration file.

.blogspot.

- All 404 are returned as `INACTIVE`
- All 301 which are blocked by Google or does not exist are returned as `INACTIVE`
- All 302 which are blocked by Google are returned as `INACTIVE`

*.canalblog.com

- All 404 are returned as `INACTIVE`

*.doubleclick.net

- All 404 are returned as `INACTIVE`

***.liveadvert.com**

- All 404 are returned as INACTIVE

***.skyrock.com**

- All 404 are returned as INACTIVE

***.tumblr.com**

- All 404 are returned as INACTIVE

***.wordpress.com**

- All 301 which match `doesn't exist` are returned as INACTIVE

IP with range

- All IPv4 with a range (for example `0.0.0.0/24`) are returned as ACTIVE

8.5 HTTP Code

Note: This section reflects exactly what the code does. So any changes made here should also be reported to the code or at least the configuration file.

Note: A `***` in this column means that it was impossible to catch the HTTP status code from the web server.

We have categorized the HTTP Code into 3 parts.

- Active
 - Consider those ones like the one that influences HTTP source logic.
 - Please note that the domain is automatically introduced into the official outputs but we keep a record of it `output/Analytic/ACTIVE`.
- Potentially Active
 - If the domain status returned by other methods are different from INACTIVE or INVALID and the HTTP status code is into that list, we save the domain into `output/Analytic/POTENTIALLY_ACTIVE`
- Inactive or potentially inactive
 - If the domain status returned by other methods are different from ACTIVE and the HTTP status code is on that list, we save the domain into `output/Analytic/POTENTIALLY_INACTIVE`

8.5.1 As active

- 100 - Continue
- 101 - Switching Protocols
- 200 - OK
- 201 - Created
- 202 - Accepted
- 203 - Non-Authoritative Information
- 204 - No Content
- 205 - Reset Content
- 206 - Partial Content

8.5.2 As potentially active

- 000
- 300 - Multiple Choices
- 301 - Moved Permanently
- 302 - Found
- 303 - See Other
- 304 - Not Modified
- 305 - Use Proxy
- 307 - Temporary Redirect
- 403 - Forbidden
- 405 - Method Not Allowed
- 406 - Not Acceptable
- 407 - Proxy Authentication Required
- 408 - Request Timeout
- 411 - Length Required
- 413 - Request Entity Too Large
- 417 - Expectation Failed
- 500 - Internal Server Error
- 501 - Not Implemented
- 502 - Bad Gateway
- 503 - Service Unavailable
- 504 - Gateway Timeout
- 505 - HTTP Version Not Supported

8.5.3 As inactive or potentially inactive

- 400 - Bad Request
- 401 - Unauthorized
- 402 - Payment Required (Not in use but may be seen in the future)
- 404 - Not Found
- 409 - Conflict
- 410 - Gone
- 412 - Precondition Failed
- 414 - Request-URI Too Long
- 415 - Unsupported Media Type
- 416 - Requested Range Not Satisfiable

9.1 Why logs sharing?

We chose to initiate the logs sharing as some actions can really be random when working with millions of domains. The idea and purpose of this feature are **ONLY** to make PyFunceble a better tool.

9.2 What do we share/collect?

Event	Shared	URL
No WHOIS server (referrer) is found.	<ul style="list-style-type: none"> The extension of the currently tested domain. 	https://pyfunceble.funilrys.com/api/no-referrer
The expiration date is not correctly formatted.	<ul style="list-style-type: none"> The extracted expiration date. The currently tested domain. The currently used WHOIS server (DNS) name. 	https://pyfunceble.funilrys.com/api/date-format

9.3 How to share logs?

The logs sharing is activated by default.

If you do not wish to share your logs simply change

```
share_logs: True
```

to

```
share_logs: False
```

into your personal *.PyFunceble.yaml*.

CHAPTER 10

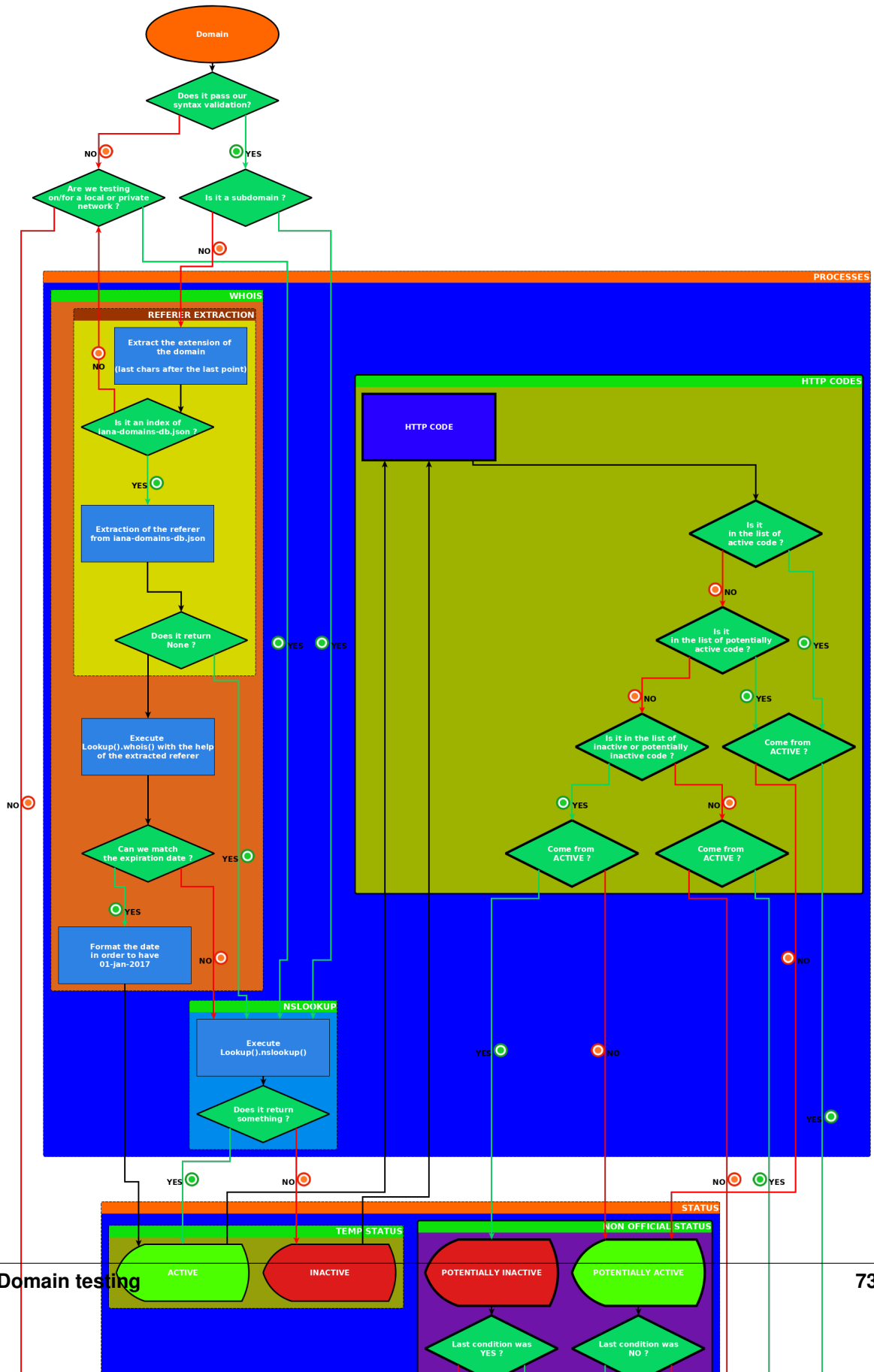
Logic representation

Don't have time to review the code and want to know exactly what PyFunceble does in general? This flowchart is there for you!

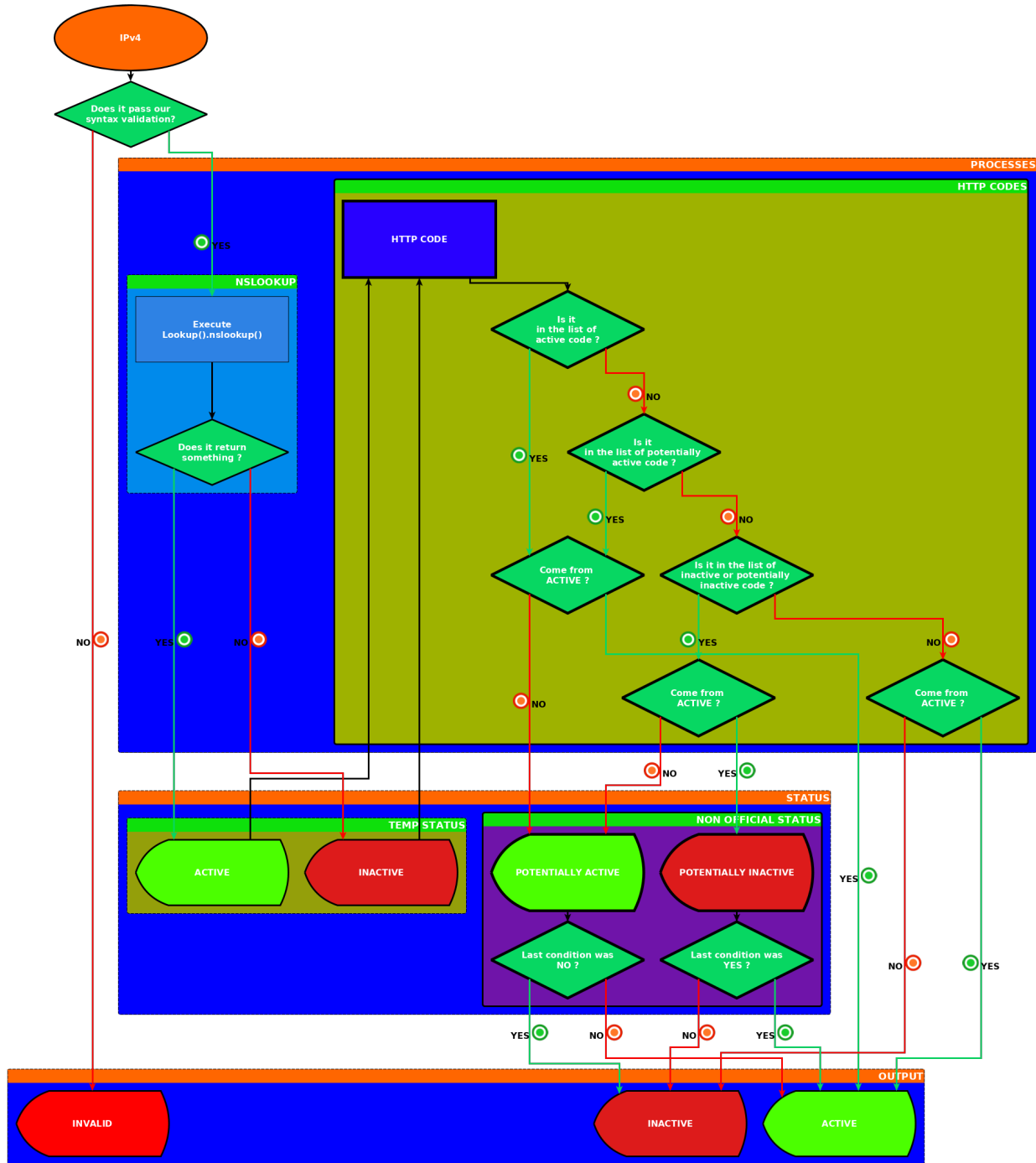
Note: This is a simplified version of what we actually do.

Note: The diagrams were generated with the help of [yEd Graph Editor](#).

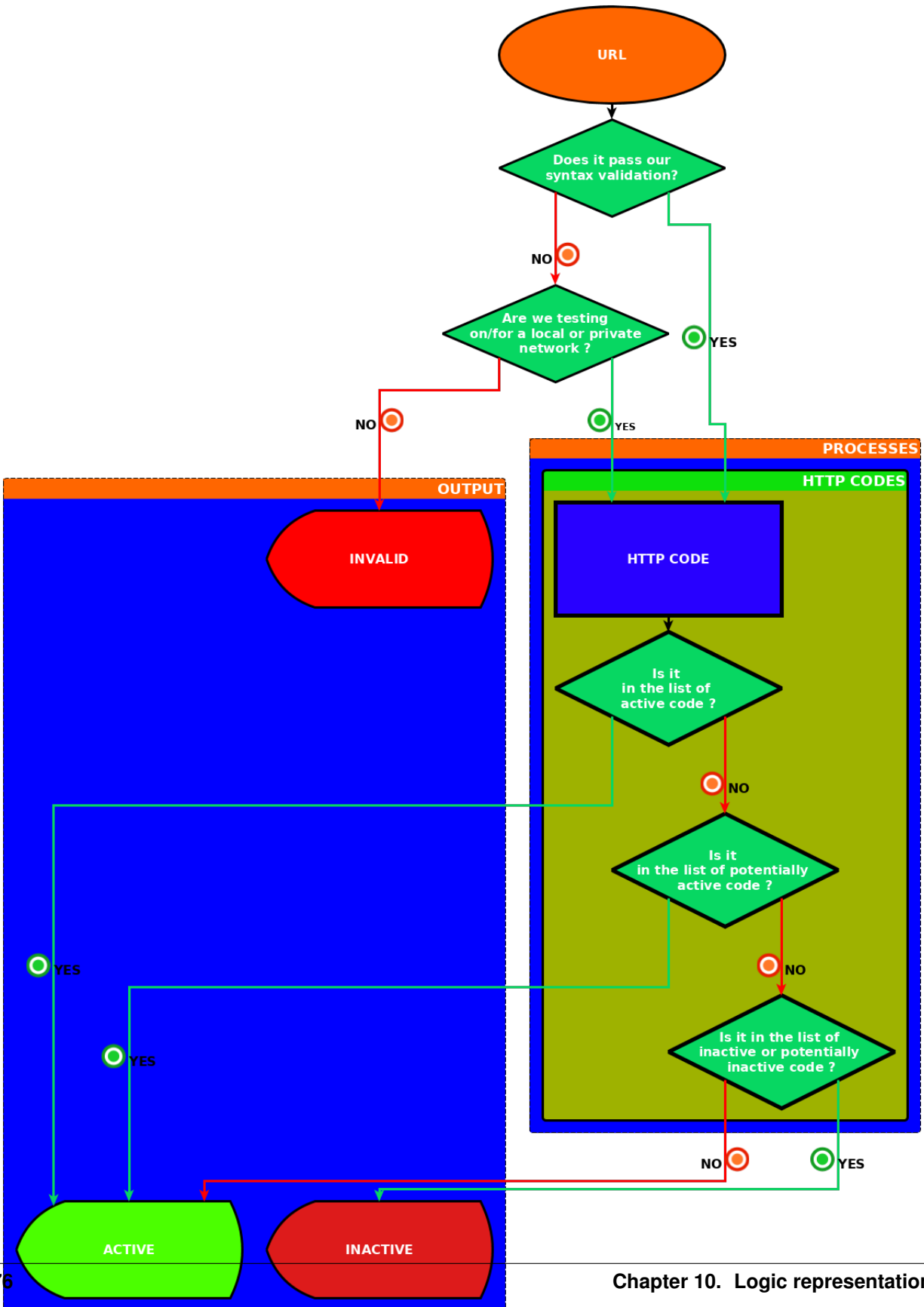
10.1 Domain testing



10.2 IP testing



10.3 URL testing



CHAPTER 11

API documentation

Note: This section document what you can call directly when you use PyFuncible as an imported module.

The tool to check the availability or syntax of domains, IPv4 or URL.

This submodule is the main entry of PyFuncible.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Special thanks: <https://pyfuncible.readthedocs.io/en/master/special-thanks.html>

Contributors: <http://pyfuncible.readthedocs.io/en/master/special-thanks.html>

Project link: <https://github.com/funilrys/PyFuncible>

Project documentation: <https://pyfuncible.readthedocs.io/en/master/>

Project homepage: <https://funilrys.github.io/PyFuncible/>

License:

MIT License

Copyright (c) 2017-2019 Nissar Chababy

Permission **is** hereby granted, free of charge, to **any** person obtaining a copy of this software **and** associated documentation files (the "**Software**"), to deal **in** the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, **and/or** sell

(continues on next page)

(continued from previous page)

copies of the Software, **and** to permit persons to whom the Software **is** furnished to do so, subject to the following conditions:

The above copyright notice **and** this permission notice shall be included **in all** copies **or** substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

`PyFuncible.ipv4_syntax_check(ip)`

Check the syntax of the given IPv4.

Parameters `ip` (*str*) – The IPv4 to check the syntax for.

Returns The syntax validity.

Return type bool

Warning: If an empty or a non-string `ip` is given, we return None.

`PyFuncible.is_ipv4_range(ip)`

Check if the given IP is an IP range.

Parameters `ip` (*str*) – The IP we are checking.

Returns The IPv4 range state.

Return type bool

Warning: If an empty or a non-string `ip` is given, we return None.

`PyFuncible.is_subdomain(domain)`

Check if the given domain is a subdomain.

Parameters `domain` (*str*) – The domain we are checking.

Returns The subdomain state.

Return type bool

Warning: If an empty or a non-string `domain` is given, we return None.

`PyFuncible.load_config(under_test=False, custom=None)`

Load the configuration.

Parameters

- **under_test** (*bool*) – Tell us if we only have to load the configuration file (True) or load the configuration file and initiate the output directory if it does not exist (False).
- **custom** (*dict*) – A dict with the configuration index (from `.PyFuncible.yaml`) to update.

Warning: If `custom` is given, the given `dict` overwrite the last value of the given configuration indexes.

PyFuncible.**syntax_check** (*domain*)

Check the syntax of the given domain.

Parameters `domain` (*str*) – The domain to check the syntax for.

Returns The syntax validity.

Return type `bool`

Warning: If an empty or a non-string `domain` is given, we return `None`.

PyFuncible.**test** (*domain*, *complete=False*, *config=None*)

Test the availability of the given domain or IP.

Parameters

- **domain** (*str*) – The domain or IP to test.
- **complete** (*bool*) – Activate the return of a `dict` with some significant data from the test.
- **config** (*dict*) – A `dict` with the configuration index (from `.PyFuncible.yaml`) to update.

Returns The status or the informations of the domain.

Return type `strdict`

Warning: If an empty or a non-string `domain` is given, we return `None`.

Warning: If `config` is given, the given `dict` overwrite the last value of the given indexes.

Note: If `complete` is set to `True`, we return the following indexes.

```
{
  "_status_source": None,
  "_status": None,
  "domain_syntax_validation": None,
  "expiration_date": None,
  "http_status_code": None,
  "ip4_syntax_validation": None,
  "nslookup": [],
  "status_source": None,
  "status": None,
  "tested": None,
  "url_syntax_validation": None,
  "whois_record": None,
  "whois_server": None,
}
```

PyFuncible.**url_syntax_check** (*url*)

Check the syntax of the given URL.

Parameters `url` (*str*) – The URL to check the syntax for.

Returns The syntax validity.

Return type `bool`

Warning: If an empty or a non-string `url` is given, we return `None`.

`PyFuncible.url_test` (*url*, *complete=False*, *config=None*)

Test the availability of the given URL.

Parameters

- `url` (*str*) – The URL to test.
- `complete` (*bool*) – Activate the return of a dict with some significant data from the test.
- `config` (*dict*) – A dict with the configuration index (from `.PyFuncible.yaml`) to update.

Returns The status or the informations of the URL.

Return type `strdict`

Warning: If an empty or a non-string `url` is given, we return `None`.

Warning: If `config` is given, the given `dict` overwrite the last value of the given indexes.

Note: This section will document every part (except the API section) of our code base.

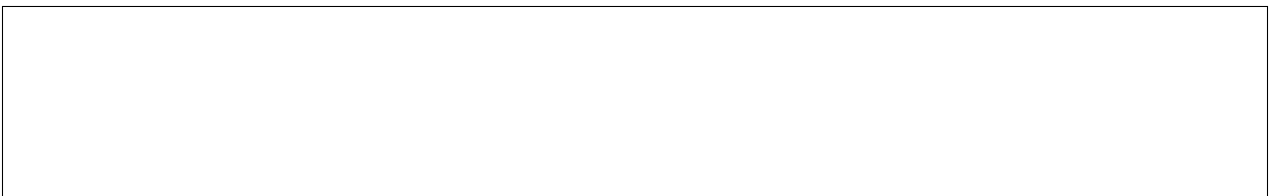
12.1 Helpers

12.1.1 Problematic

How can we avoid writing the same thing every time?

12.1.2 Documentation

The tool to check the availability or syntax of domains, IPv4 or URL.



This submodule will provide the helpers.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Special thanks: <https://pyfunceble.readthedocs.io/en/master/special-thanks.html>

Contributors: <http://pyfunceble.readthedocs.io/en/master/special-thanks.html>

Project link: <https://github.com/funilrys/PyFunceble>

Project documentation: <https://pyfunceble.readthedocs.io/en/master/>

Project homepage: <https://funilrys.github.io/PyFunceble/>

License:

```
MIT License

Copyright (c) 2017-2019 Nissar Chababy

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

class `PyFunceble.helpers.Command` (*command*)

Shell command execution.

Parameters `command` (*str*) – The command to execute

`_decode_output` (*to_decode*)

Decode the output of a shell command in order to be readable.

Parameters `to_decode` – Output of a command to decode.

Type bytes

Returns The decoded output.

Return type str

`execute` ()

Execute the given command.

Returns The output of the command.

Return type str

`run` ()

Run the given command and yield each line(s) one by one.

Note: The difference between this method and `self.execute()` is that `self.execute()` wait for the process to end in order to return its output.

class `PyFunceble.helpers.Dict` (*main_dictionnaire=None*)

Dictionary manipulations.

Parameters `main_dictionnaire` (*dict*) – The dict we are working with.

classmethod `from_json` (*data*)

Convert a JSON formatted string into a dictionary.

Parameters `data` (*str*) – A JSON formatted string to convert to dict format.

Returns The dict representation of the JSON formatted string.

Return type dict

classmethod `from_yaml` (*data*)

Convert a YAML formatted string into a dictionary.

Parameters `data` (*str*) – A YAML formatted string to convert to dict format.

Returns The dict representation of the YAML formatted string.

Return type dict

merge (*to_merge*, *strict=True*)

Merge the content of `to_merge` into the given main dictionary.

Parameters

- **to_merge** (*dict*) – The dictionary to merge.
- **strict** (*bool*) – Tell us if we have to strictly merge lists.
True: We follow index :code‘False‘: We follow element (content)

Returns The merged dict.

Return type dict

remove_key (*key_to_remove*)

Remove a given key from a given dictionary.

Parameters `key_to_remove` (*list/str*) – The key(s) to delete.

Returns The dict without the given key(s).

Return type dict|None

rename_key (*key_to_rename*, *strict=True*)

Rename the given keys from the given dictionary.

Parameters

- **key_to_rename** (*dict*) – The key(s) to rename. Expected format: {old:new}
- **strict** – Tell us if we have to rename the exact index or the index which looks like the given key(s)

Returns The well formatted dict.

Return type dict|None

to_json (*destination*)

Save a dictionary into a JSON file.

Parameters `destination` (*str*) – A path to a file where we’re going to write the converted dict into a JSON format.

to_yaml (*destination*, *flow_style=False*)

Save a dictionary into a YAML file.

Parameters `destination` (*str*) – A path to a file where we’re going to write the converted dict into a JSON format.

class `PyFuncible.helpers.Directory` (*directory*)

Directory manipulation.

Parameters `directory` (*str*) – A path to the directory to manipulate.

fix_path (*splited_path=None*)

Fix the path of the given path.

Parameters `splited_path` (*list*) – A list to convert to the right path.

Returns The fixed path.

Return type `str`

class `PyFuncible.helpers.Download` (*link*, *destination=None*, *return_data=False*, *verify_certificate=True*)

Download or return the content of the given link.

Parameters

- **link** (*str*) – The link to download.
- **destination** (*str*) – The location where we should save the downloaded content.
- **return_data** (*bool*) – Tell us if we need to return the page content or write its content into the given destination.
- **verify_certificate** (*bool*) – Tell us if we need to verify the SSL/TLS certificate.

text ()

Download the given link and return or save its `requests.text` at the given destination.

Return type `mixed`

Raises

Exception If the status code is not 200.

class `PyFuncible.helpers.File` (*file*)

File treatment/manipulations.

Parameters `file` (*str*) – A path to the file to manipulate.

copy (*destination*)

Copy the given file to the destination.

Parameters `destination` (*str*) – The destination of the copy.

delete ()

Delete a given file path.

read ()

Read a given file path and return its content.

Returns The content of the given file path.

Return type `str`

write (*data_to_write*, *overwrite=False*)

Write or append data into the given file path.

Parameters

- **data_to_write** (*str*) – The data to write.
- **overwrite** (*bool*) – Tell us if we have to overwrite the content of the file we are working with.

class `PyFuncible.helpers.Hash` (*file_path=None*, *algorithm='sha512'*, *only_hash=False*, *data=None*)
 Get and return the hash a file with the given algorithm.

Parameters

- **file_path** (*str*) – The path to the file we have to hash.
- **algorithm** (*str*) – The algorithm to use.
- **only_hash** – Tell us if we only have to return the desired algorithm instead of the dummy dict format.

Note: The original version can be found at <https://git.io/vFQrK>.

_hash_data (*algo*)

Get hash of the given data.

Parameters **algo** (*str*) – The algorithm to use.

_hash_file (*algo*)

Get the hash of the given file

Parameters **algo** (*str*) – The algorithm to use.

Returns The hexdigest of the data.

Return type `str`

get ()

Return the hash of the given file

class `PyFuncible.helpers.List` (*main_list=None*)

List manipulation.

Parameters **main_list** (*list*) – The list to manipulate.

custom_format (*key_method*, *reverse=False*)

Return a well formatted list. With the *key_method* as a function/method to format the elements before sorting.

Parameters

- **key_method** (*function/method*) – A function or method to use to format the readed element before sorting.
- **reverse** (*bool*) – Tell us if we have to reverse the list.

Returns A sorted list.

Return type `list`

format ()

Return a well formatted list. Basicaly, it's sort a list and remove duplicate.

Returns A sorted, without duplicate, list.

Return type `list`

merge (*to_merge*, *strict=True*)

Merge *to_merge* into the given main list.

Parameters

- **to_merge** (*list*) – The list to merge.

- **strict** (*bool*) – Tell us if we have to respect index (True) or not (False).

Returns The merged list.

Return type list

class `PyFunceble.helpers.Regex` (*data, regex, **args*)

A simple implementation of the python.re package

Parameters

- **data** (*str*) – The data to check.
- **regex** (*str*) – The regex to match.
- **group** (*int*) – The group to return.
- **rematch** (*bool*) – Allow to return the matched groups into a formatted list.

Note: This is an implementation of Bash `${BASH_REMATCH}`

- **replace_with** (*str*) – The value to replace the matched regex with.
- **occurrences** (*int*) – The number of occurrence(s) to replace.
- **return_type** – Tell us if we have to return the matched data or simply check if we matched (True) or not (False)

match ()

Used to get exploitable result of re.search

Returns The data of the match status.

Return type mixed

matching_list ()

Return a list of the string which match the given regex.

not_matching_list ()

Return a list of string which don't match the given regex.

replace ()

Used to replace a matched string with another.

Returns The data after replacement.

Return type str

Download ()

class `PyFunceble.helpers.Download` (*link, destination=None, return_data=False, verify_certificate=True*)

Download or return the content of the given link.

Parameters

- **link** (*str*) – The link to download.
- **destination** (*str*) – The location where we should save the downloaded content.
- **return_data** (*bool*) – Tell us if we need to return the page content or write its content into the given destination.
- **verify_certificate** (*bool*) – Tell us if we need to verify the SSL/TLS certificate.

text ()

Download the given link and return or save its `requests.text` at the given destination.

Return type mixed

Raises

Exception If the status code is not 200.

Command ()

class `PyFuncible.helpers.Command (command)`

Shell command execution.

Parameters `command (str)` – The command to execute

`_decode_output (to_decode)`

Decode the output of a shell command in order to be readable.

Parameters `to_decode` – Output of a command to decode.

Type bytes

Returns The decoded output.

Return type str

execute ()

Execute the given command.

Returns The output of the command.

Return type str

run ()

Run the given command and yield each line(s) one by one.

Note: The difference between this method and `self.execute ()` is that `self.execute ()` wait for the process to end in order to return its output.

Regex ()

class `PyFuncible.helpers.Regex (data, regex, **args)`

A simple implementation of the `python.re` package

Parameters

- **data (str)** – The data to check.
- **regex (str)** – The regex to match.
- **group (int)** – The group to return.
- **rematch (bool)** – Allow to return the matched groups into a formatted list.

Note: This is an implementation of Bash `${BASH_REMATCH}`

- **replace_with (str)** – The value to replace the matched regex with.
- **occurrences (int)** – The number of occurrence(s) to replace.

- **return_type** – Tell us if we have to return the matched data or simply check if we matched (True) or not (False)

match()

Used to get exploitable result of re.search

Returns The data of the match status.

Return type mixed

matching_list()

Return a list of the string which match the given regex.

not_matching_list()

Return a list of string which don't match the given regex.

replace()

Used to replace a matched string with another.

Returns The data after replacement.

Return type str

Dict()

class PyFunceble.helpers.Dict (*main_dictionary=None*)

Dictionary manipulations.

Parameters **main_dictionary** (*dict*) – The dict we are working with.

classmethod **from_json** (*data*)

Convert a JSON formatted string into a dictionary.

Parameters **data** (*str*) – A JSON formatted string to convert to dict format.

Returns The dict representation of the JSON formatted string.

Return type dict

classmethod **from_yaml** (*data*)

Convert a YAML formatted string into a dictionary.

Parameters **data** (*str*) – A YAML formatted string to convert to dict format.

Returns The dict representation of the YAML formatted string.

Return type dict

merge (*to_merge, strict=True*)

Merge the content of to_merge into the given main dictionary.

Parameters

- **to_merge** (*dict*) – The dictionary to merge.
- **strict** (*bool*) – Tell us if we have to strictly merge lists.
True: We follow index :code'False': We follow element (content)

Returns The merged dict.

Return type dict

remove_key (*key_to_remove*)

Remove a given key from a given dictionary.

Parameters `key_to_remove` (*list/str*) – The key(s) to delete.

Returns The dict without the given key(s).

Return type dict|None

rename_key (*key_to_rename, strict=True*)

Rename the given keys from the given dictionary.

Parameters

- **key_to_rename** (*dict*) – The key(s) to rename. Expected format: {old:new}
- **strict** – Tell us if we have to rename the exact index or the index which looks like the given key(s)

Returns The well formatted dict.

Return type dict|None

to_json (*destination*)

Save a dictionary into a JSON file.

Parameters **destination** (*str*) – A path to a file where we’re going to write the converted dict into a JSON format.

to_yaml (*destination, flow_style=False*)

Save a dictionary into a YAML file.

Parameters **destination** (*str*) – A path to a file where we’re going to write the converted dict into a JSON format.

List()

class PyFunceble.helpers.**List** (*main_list=None*)

List manipulation.

Parameters **main_list** (*list*) – The list to manipulate.

custom_format (*key_method, reverse=False*)

Return a well formatted list. With the `key_method` as a function/method to format the elements before sorting.

Parameters

- **key_method** (*function/method*) – A function or method to use to format the readed element before sorting.
- **reverse** (*bool*) – Tell us if we have to reverse the list.

Returns A sorted list.

Return type list

format()

Return a well formatted list. Basically, it’s sort a list and remove duplicate.

Returns A sorted, without duplicate, list.

Return type list

merge (*to_merge, strict=True*)

Merge `to_merge` into the given main list.

Parameters

- **to_merge** (*list*) – The list to merge.
- **strict** (*bool*) – Tell us if we have to respect index (True) or not (False).

Returns The merged list.

Return type list

Directory()

class PyFunceble.helpers.**Directory** (*directory*)
Directory manipulation.

Parameters **directory** (*str*) – A path to the directory to manipulate.

fix_path (*splited_path=None*)
Fix the path of the given path.

Parameters **splited_path** (*list*) – A list to convert to the right path.

Returns The fixed path.

Return type str

File()

class PyFunceble.helpers.**File** (*file*)
File treatment/manipulations.

Parameters **file** (*str*) – A path to the file to manipulate.

copy (*destination*)
Copy the given file to the destination.

Parameters **destination** (*str*) – The destination of the copy.

delete ()
Delete a given file path.

read ()
Read a given file path and return its content.

Returns The content of the given file path.

Return type str

write (*data_to_write, overwrite=False*)
Write or append data into the given file path.

Parameters

- **data_to_write** (*str*) – The data to write.
- **overwrite** (*bool*) – Tell us if we have to overwrite the content of the file we are working with.

Hash()

class PyFunceble.helpers.**Hash** (*file_path=None, algorithm='sha512', only_hash=False, data=None*)
Get and return the hash a file with the given algorithm.

Parameters

- **file_path** (*str*) – The path to the file we have to hash.
- **algorithm** (*str*) – The algorithm to use.
- **only_hash** – Tell us if we only have to return the desired algorithm instead of the dummy dict format.

Note: The original version can be found at <https://git.io/vFQrK>.

_hash_data (*algo*)

Get hash of the given data.

Parameters **algo** (*str*) – The algorithm to use.

_hash_file (*algo*)

Get the hash of the given file

Parameters **algo** (*str*) – The algorithm to use.

Returns The hexdigest of the data.

Return type str

get ()

Return the hash of the given file

12.2 AdBlock

12.2.1 Problematic

How can we efficiently decode AdBlock filter list?

12.2.2 Documentation

class PyFuncible.adblock.**AdBlock** (*list_from_file, aggressive=False*)
Provide the adblock decoding logic.

Parameters **list_from_file** (*list*) – The file in list format.

_extract_base (*element*)

Extract the base of the given element.

Parameters **element** (*str/list*) – The element we are working with.

_format_decoded (*to_format, result=None*)

Format the extracted adblock line before passing it to the system.

Parameters

- **to_format** (*str*) – The extracted line from the file.
- **result** (*list*) – A list of the result of this method.

Returns The list of domains or IP to test.

Return type list

`_handle_options` (*options*)

Handle the data from the options.

Parameters `options` (*list*) – The list of options from the rule.

Returns The list of domains to return globally.

Return type list

classmethod `_is_to_ignore` (*line*)

Check if we have to ignore the given line.

Parameters `line` (*str*) – The line from the file.

`_remove_ignored` (*list_from_file*)

Removed the ignored element from the given list.

Parameters `list_from_file` – The list which represent the file we are decoding.

Returns The filtered list.

Return type list

`decode` ()

Decode/extract the domains to test from the adblock formatted file.

Returns The list of domains to test.

Return type list

12.3 Auto-continue

12.3.1 Problematic

How can we continue the test after executable stop?

12.3.2 Documentation

class `PyFunceble.auto_continue.AutoContinue`

Provide the auto-continue subsystem.

backup ()

Backup the current execution state.

restore ()

Restore data from the given path.

12.4 Auto-save

Note: Only Travis CI is actually supported.

12.4.1 Travis CI problematic

How can we bypass the default Travis CI timeout of 45 minutes?

12.4.2 Documentation

class `PyFuncible.auto_save.AutoSave` (*is_last_domain=False, is_bypass=False*)
Provide the autosave logic.

Parameters

- **is_last_domain** (*bool*) – Tell this subsystem if we are at the very end of the file testing.
- **is_bypass** (*bool*) – Tell this subsystem if we are in bypassing mode.

__travis ()

Logic behind autosave under Travis CI.

classmethod **travis_permissions** ()

Set permissions in order to avoid issues before committing.

12.5 Check

12.5.1 Problematic

How can we efficiently check the format of IP, domains, and URL?

12.5.2 Documentation

class `PyFuncible.check.Check` (*element=None*)
Provide a place to check several things around URL, IP or domain.

Parameters **element** (*str*) – The element (URL, IP or domain) to check.

is_domain_valid (*domain=None, subdomain_check=False*)

Check if the given domain is a valid.

Parameters

- **domain** (*str*) – The domain to validate.
- **subdomain_check** (*bool*) – Activate the subdomain checking.

Returns The validity of the sub-domain.

Return type `bool`

is_ip_range (*ip_to_check=None*)

Check if the given IP is a valid IPv4.

Parameters **ip_to_check** (*str*) – The IP to test.

Returns The validity of the IP.

Return type `bool`

Note: We only test IPv4 because for now we only them for now.

is_ip_valid (*ip_to_check=None*)

Check if the given IP is a valid IPv4.

Parameters **ip_to_check** (*str*) – The IP to test.

Returns The validity of the IP.

Return type bool

Note: We only test IPv4 because for now we only them for now.

is_subdomain (*domain=None*)

Check if the given subdomain is a subdomain.

Parameters **domain** (*str*) – The domain to validate.

Returns The validity of the subdomain.

Return type bool

is_url_valid (*url=None, return_base=False, return_formatted=False*)

Check if the given URL is valid.

Parameters

- **url** (*str*) – The url to validate.
- **return_base** – Allow us the return of the url base (if URL formatted correctly).
- **return_formatted** (*bool*) – Allow us to get the URL converted to IDNA if the conversion is activated.

Returns The validity of the URL or its base.

Return type boolstr

12.6 Cleaning

12.6.1 Problematic

How can we clean the `output/` directory so we do not have a collision between old and new files?

12.6.2 Documentation

class `PyFunceble.clean.Clean` (*list_to_test, clean_all=False*)

Provide the cleaning logic(s).

Parameters

- **list_to_test** (*list|None*) – The list of domains we are testing.
- **clean_all** (*bool*) – Tell the subsystem if we need to clean all. Which include, of course, the output directory but also all other file(s) generated by our system.

12.7 Configuration

12.7.1 Problematics

- How can we avoid the usage of `tool.py`?
- How can we make personalization more simple?

12.7.2 Documentation

Load()

class `PyFuncible.config.Load(path_to_config)`

Help us load the configuration(s) file(s).

Parameters `path_to_config` (*str*) – The possible path to the configuration to load.

classmethod `_install_directory_structure_file()`

Download the latest version of `dir_structure_production.json`.

classmethod `_install_iana_config()`

Download `iana-domains-db.json` if not present.

classmethod `_install_production_config()`

Download the production configuration and install it in the current directory.

classmethod `_install_psl_config()`

Download `public-suffix.json` if not present.

classmethod `_load_config_file()`

Load `.PyFuncible.yaml` into the system.

classmethod `_set_path_to_configs(path_to_config)`

Set the paths to the configuration files.

Parameters `path_to_config` (*str*) – The possible path to the config to load.

Returns The path to the config to read (0), the path to the default configuration to read as fall-back.(1)

Return type tuple

Version()

class `PyFuncible.config.Version(used=False)`

Compare the local with the upstream version.

Parameters `used` (*bool*) – True: Version is configured for simple usage. False: Version compare local with upstream.

classmethod `check_versions(local, upstream)`

Compare the given versions.

Parameters

- **local** (*list*) – The local version converted by `split_versions()`.
- **upstream** (*list*) – The upstream version converted by `split_versions()`.

Returns

- True: local < upstream
- None: local == upstream
- False: local > upstream

Return type bool|None

classmethod `check_versions_literally(local, upstream)`

Compare the given versions literally.

Parameters

- **local** (*str*) – The local version converted by `split_versions()`.
- **upstream** (*str*) – The upstream version converted by `split_versions()`.

Returns

- True: local == upstream
- False: local != upstream

Return type bool

compare()

Compare the current version with the upstream saved version.

classmethod is_cloned()

Let us know if we are currently in the cloned version of PyFuncible which implicitly mean that we are in development mode.

classmethod right_url_from_version(url)

Convert the GitHub URL to the right one depending of the branch or version we are working with.

Parameters **url** (*str*) – The URL to convert.

Returns The converted URL.

Return type str

classmethod split_versions(version, return_non_digits=False)

Convert the versions to a shorter one.

Parameters

- **version** (*str*) – The version to split.
- **return_non_digits** (*bool*) – Activate the return of the non-digits parts of the splitted version.

Returns The splitted version name/numbers.

Return type list

12.8 Core

12.8.1 Documentation

class `PyFuncible.core.Core` (**args)

Main entry to PyFuncible. Brain of the program. Also known as “put everything together to make the system works”.

Parameters

- **domain_or_ip_to_test** (*str*) – A domain or IP to test.
- **file_path** (*str*) – A path to a file to read and test.
- **url_to_test** (*str*) – A URL to test.
- **url_file** (*str*) – A path to a file which contains URL to test.
- **link_to_test** (*str*) – A link to a file to download and test.

- **modulo_test** – If set to True, it will tell the system that we are working as an exported module.
- **modulo_test** – bool

__entry_management ()

Avoid to have 1 millions line into self.__init__()

__entry_management_url ()

Manage the loading of the url system.

__entry_management_url_download (*passed*)

Check if the given information is a URL. If it is the case, it download and update the location of file to test.

Parameters **passed** (*str*) – The url passed to the system.

Returns The state of the check.

Return type bool

classmethod **__extract_domain_from_file** ()

Extract all non commented lines from the file we are testing.

Returns The elements to test.

Return type list

__file_decision (*current, last, status=None*)

Manage the database, autosave and autocontinue systems for the case that we are reading a file.

Parameters

- **current** (*str*) – The currently tested element.
- **last** (*str*) – The last element of the list.
- **status** (*str*) – The status of the currently tested element.

__file_list_to_test_filtering ()

Unify the way we work before testing file contents.

classmethod **__format_domain** (*extracted_domain*)

Format the extracted domain before passing it to the system.

Parameters **extracted_domain** (*str*) – The extracted domain.

Returns The formatted domain or IP to test.

Return type str

classmethod **__print_header** ()

Decide if we print or not the header.

classmethod **bypass** ()

Exit the script if [PyFuncible skip] is matched into the latest commit message.

classmethod **colorify_logo** (*home=False*)

Print the colored logo based on global results.

Parameters **home** (*bool*) – Tell us if we have to print the initial coloration.

domain (*domain=None, last_domain=None*)

Manage the case that we want to test only a domain.

Parameters

- **domain** (*str*) – The domain or IP to test.

- **last_domain** (*str*) – The last domain to test if we are testing a file.
- **return_status** (*bool*) – Tell us if we need to return the status.

file ()

Manage the case that need to test each domain of a given file path.

Note: 1 domain per line.

file_url ()

Manage the case that we have to test a file

Note: 1 URL per line.

classmethod reset_counters ()

Reset the counters when needed.

classmethod switch (*variable, custom=False*)

Switch PyFuncible.CONFIGURATION variables to their opposite.

Parameters

- **variable** (*str/bool*) – The variable name to switch. The variable should be an index our configuration system. If we want to switch a bool variable, we should parse it here.
- **custom** (*bool*) – Let us know if have to switch the parsed variable instead of our configuration index.

Returns The opposite of the configuration index or the given variable.

Return type bool

Raises

Exception When the configuration is not valid. In other words, if the PyFuncible.CONFIGURATION[variable_name] is not a bool.

test (*complete=False*)

Avoid confusion between self.domain which is called into __main__ and test() which should be called out of PyFuncible's scope.

Parameters complete (*bool*) – Activate the return of a dictionary with signigican data about the test.

Returns ACTIVE INACTIVE or INVALID.

Return type strllist

Raises

Exception When this method is called under __name__ == '__main__'

Note: This method should never be called in a __name__ == '__main__' context.

test_with_complete_information ()

Run a test and return all available informations.

Note: The following are the indexes which we return. Please report to the advanced usage documentation for a description of each indexes.

```
{
  "whois_server": None,
  "whois_record": None,
  "url_syntax_validation": None,
  "tested": None,
  "status": None,
  "status_source": None,
  "nslookup": [],
  "ip4_syntax_validation": None,
  "http_status_code": None,
  "expiration_date": None,
  "domain_syntax_validation": None,
  "_status": None,
  "_status_source": None,
}
```

url (*url_to_test=None, last_url=None*)

Manage the case that we want to test only a given url.

Parameters

- **url_to_test** (*str*) – The url to test.
- **last_url** (*str*) – The last url of the file we are testing (if exist)

12.9 Database

12.9.1 Problematics

- How can we continuously test INACTIVE and INVALID domains or IP?
- How can we reduce the number of whois requests over time?

12.9.2 Documentation

class `PyFuncible.database.Inactive`

Logic behind the generation and the usage of a database system. The main idea behind this is to provide an inactive-db.json and test all inactive domain which are into to it regularly

`_add_to_test` (*to_add*)

Add an element or a list of element into `PyFuncible.INTERN['inactive_db'] [PyFuncible.INTERN["file_to_test"]]['to_test']`.

Parameters `to_add` (*str/list*) – The domain, IP or URL to add.

`_backup` ()

Save the current database into the inactive-db.json file.

`_merge` ()

Merge the real database with the older one which has already been set into `PyFuncible.INTERN["inactive_db"]`

`_reformat_historical_formatting_error()`

Format the old format so it can be merged into the newer format.

`_retrieve()`

Return the current content of the inactive-db.json file.

`_timestamp()`

Get the timestamp where we are going to save our current list.

Returns The timestamp to append with the currently tested element.

Return type int|str

`add()`

Save the current :code: 'PyFuncible.CONFIGURATION['to_test']' into the current timestamp.

`classmethod content()`

Get the content of the database.

Returns The content of the database.

Return type list

`classmethod is_present()`

Check if the currently tested element is into the database.

`remove()`

Remove all occurrence of `PyFuncible.CONFIGURATION['to_test']` from the database.

`to_test()`

Get the list to test for the next session.

`class PyFuncible.database.whois(expiration_date=None)`

Logic behind the whois database. Indeed, the idea is to implement #2.

Parameters `expiration_date` (*str*) – The extracted expiration date.

`classmethod _authorization()`

Check if we are authorized to work with our database.

`_backup()`

Backup the database into its file.

`_retrieve()`

Retrieve the data from the database.

`add()`

Add the currently tested element into the database.

`get_expiration_date()`

Get the expiration date from the database.

Returns The expiration date from the database.

Return type str|None

`is_in_database()`

Check if the element is into the database.

`is_time_older()`

Check if the current time is older than the one in the database.

12.10 Directory Structure

12.10.1 Problematic

How can we give make the output directory less **annoying** after an update?

12.10.2 Documentation

class `PyFuncible.directory_structure.DirectoryStructure` (*production=False*)

Basically a backup/reconstructor of our output directory.

Parameters `production` (*bool*) – Tell the subsystem if we are preparing for production which imply the execution of the backup insteam of of the “reconstructore” mode.

classmethod `_create_directory` (*directory, loop=False*)

Creates the given directory if it does not exists.

Parameters

- `directory` (*str*) – The directory to create.
- `loop` (*bool*) – Tell us if we are in the creation loop or not.

`_get_structure` ()

Get the structure we are going to work with.

Returns The structure we have to work with.

Return type dict

`_restore_replace` ()

Check if we need to replace “.gitignore” to “.keep”.

Returns The replacement status.

Return type bool

`_update_structure_from_config` (*structure*)

Update the paths according to configs.

Parameters `structure` (*dict*) – The read structure.

`backup` ()

Backup the developer state of *output/* in order to make it restorable and portable for user.

`delete_unneeded` ()

Delete the directory which are not registered into our structure.

`restore` ()

Restore the ‘output/’ directory structure based on the *dir_structure.json* file.

12.11 Execution Time

12.11.1 Problematic

How to monitor the execution time of the session?

12.11.2 Documentation

class `PyFuncible.execution_time.ExecutionTime` (*action='start', last=False*)
Set and return the execution time of the program.

Parameters

- **action** (*str*) – The action related the execution time. Can be *start* or *stop*.
- **last** (*bool*) – Tell the subsystem if we are at the very end of the file testing.

classmethod `_authorization` ()

Check the execution authorization.

Returns The authorization status.

Return type `bool`

classmethod `_calculate` (*start=None, end=None*)

calculate the difference between starting and ending time.

Parameters

- **start** (*int / str*) – A starting time.
- **stop** (*int / str*) – A ending time.

Returns

A dict with following as index.

- `days`
- `hours`
- `minutes`
- `seconds`

as index.

Return type `dict`

method `_save` (*last=False*)

Save the current time to the file.

Parameters **last** (*bool*) – Tell us if we are at the very end of the file testing.

classmethod `_starting_time` ()

Set the starting time.

classmethod `_stopping_time` ()

Set the ending time.

format_execution_time (*start=None, end=None*)

Format the calculated time into a human readable format.

Parameters

- **start** (*int / str*) – A starting time.
- **stop** (*int / str*) – A ending time.

Returns A human readable date.

Return type `str`

12.12 Expiration Date

12.12.1 Problematic

How can we get the expiration date of a given domain?

12.12.2 Documentation

class `PyFuncible.expiration_date.ExpirationDate`

Get, format and return the expiration date of a domain, if exist.

`_cases_management` (*regex_number, matched_result*)

A little internal helper of `self.format`. (Avoiding of nested loops)

Note: Please note that the second value of the case represent the groups in order [*day, month, year*].

This means that a [*2, 1, 0*] will be for example for a date in format 2017-01-02 where 01 is the month.

Parameters

- **`regex_number`** (*int*) – The identifiant of the regex.
- **`matched_result`** (*list*) – The matched result to format.

Returns A list representing the expiration date. The list can be “decoded” like [*day, month, year*]

Return type list|None

classmethod `_convert_1_to_2_digits` (*number*)

Convert 1 digit number to two digits.

Parameters **`number`** (*str/int*) – A number or a digit string.

Returns A 2 or more digit string.

Return type str

classmethod `_convert_or_shorten_month` (*data*)

Convert a given month into our unified format.

Parameters **`data`** (*str*) – The month to convert or shorten.

Returns The unified month name.

Return type str

`_extract` ()

Extract the expiration date from the whois record.

Returns The status of the domain.

Return type str

`_format` (*date_to_convert=None*)

Format the expiration date into an unified format (01-jan-1970).

Parameters **`date_to_convert`** (*str*) – The date to convert. In other words, the extracted date.

Returns The formatted expiration date.

Return type str

get ()

Execute the logic behind the meaning of ExpirationDate + return the matched status.

Returns The status of the tested domain. Can be one of the official status.

Return type str

12.13 Generation

12.13.1 Problematic

How can we generate files which reflects the results of PyFuncible?

12.13.2 Documentation

class PyFuncible.generate.**Generate** (*domain_status*, *source=None*, *expiration_date=None*)

Generate different sort of files.

Parameters

- **domain_status** (*str*) – The domain status.
- **source** (*str*) – The source of the given status.
- **expiration_date** (*str*) – The expiration date of the domain (if caught).

_analytic_host_file_directory ()

Return the analytic directory to write depending of the matched status.

_do_not_produce_file ()

Check if we are allowed to produce a file based from the given information.

Returns The state of the production. True: We do not produce file. False: We do produce file.

Return type bool

classmethod **_handle_non_existant_index** ()

Handle and check that some configuration index exists.

_prints_status_file ()

Logic behind the printing (in file) when generating status file.

_prints_status_screen ()

Logic behind the printing (on screen) when generating status file.

analytic_file (*new_status*, *old_status=None*)

Generate Analytic/* files based on the given old and new statuses.

Parameters

- **new_status** (*str*) – The new status of the domain.
- **old_status** (*str*) – The old status of the domain.

info_files ()

Generate the hosts file, the plain list and the splitted lists.

status_file()

Generate a file according to the domain status.

unified_file()

Generate unified file. Understand by that that we use an unified table instead of a separate table for each status which could result into a misunderstanding.

12.14 HTTP Code

12.14.1 Problematic

How can we get the HTTP status code of the given domain or IP?

12.14.2 Documentation

class `PyFunceble.http_code.HTTPCode`

Get and return the HTTP code status of a given domain.

_access()

Get the HTTP code status.

Returns The matched HTTP status code.

Return type `int|None`

get()

Return the HTTP code status.

Returns The matched and formatted status code.

Return type `str|int|None`

12.15 IANA

12.15.1 Problematic

How can we get information from IANA?

12.15.2 Documentation

class `PyFunceble.iana.IANA`

Logic behind the update and usage of *iana-domains-db.json*

_extensions()

Extract the extension from the given block. Plus get its referer.

_referer(extension)

Return the referer for the given extension.

Parameters **extension** (*str*) – A valid domain extension.

Returns The whois server to use to get the WHOIS record.

Return type `str`

load()

Initiate the IANA database if it is not the case.

update()

Update the content of the *iana-domains-db* file.

12.16 Logs

12.16.1 Problematic

How can we efficiently generate and share logs?

12.16.2 Documentation

class `PyFuncible.logs.Logs` (*output=None*)

Provide a clean and unique way to work with logs. Indeed, it's not good to have logs spread around the code :smile:

Parameters `output` (*str*) – A path to the JSON file we are going to write.

classmethod `_get_content` (*file*)

Get and return the content of the given log file.

Parameters `file` (*str*) – The file we have to get the content from.

:return The content of the given file. :rtype: dict

classmethod `_write_content` (*content, file*)

Write the content into the given file.

Parameters

- **content** (*dict*) – The dict to write.
- **file** (*str*) – The file to write.

expiration_date (*extracted*)

Logs the extracted expiration date.

Parameters `extracted` (*str*) – The extracted expiration date (from WHOIS record).

referer_not_found (*extension*)

Logs the case that the referer was not found.

Parameters `extension` (*str*) – The extension of the domain we are testing.

whois (*record*)

Logs the WHOIS record if needed.

Parameters `record` (*str*) – The record to log.

12.17 Lookup

12.17.1 Problematics

- How can we get information from WHOIS records?

- How can we check if a domain or IP have a valid pointer (nslookup)?

12.17.2 Documentation

class `PyFunceble.lookup.Lookup`

Can be used to NSLOOKUP or WHOIS lookup.

classmethod `nslookup()`

Implementation of UNIX nslookup.

classmethod `whois(whois_server, domain=None, timeout=None)`

Implementation of UNIX whois.

Parameters

- **whois_server** (*str*) – The WHOIS server to use to get the record.
- **domain** (*str*) – The domain to get the whois record from.
- **timeout** (*int*) – The timeout to apply to the request.

Returns The whois record from the given whois server, if exist.

Return type `str|None`

12.18 Mining

12.18.1 Problematic

How can we get the list of domain or URL which link to the desired domain, IPv4 or URL?

12.18.2 Documentation

class `PyFunceble.mining.Mining`

Manage the minig subsystem.

_add (*to_add*)

Add the currently mined information to the mined “database”.

Parameters `to_add(dict)` – The element to add.

_backup ()

Backup the mined informations.

_retrieve ()

Retrieve the mining informations.

classmethod `list_of_mined()`

Provide the list of mined so they can be added to the list queue.

Returns The list of mined domains or URL.

Return type `list`

mine ()

Search for domain or URL related to the original URL or domain.

Returns The mined domains or URL.

Return type dict

process ()

Process the logic and structuration of the mining database.

remove ()

Remove the currently tested element from the mining data.

12.19 Percentage

12.19.1 Problematic

How can we calculate the percentage of each status?

12.19.2 Documentation

class PyFunceble.percentage.**Percentage** (*domain_status=None, init=None*)

Calculation of the percentage of each status.

Parameters

- **domain_status** (*str*) – The status to increment.
- **init** (*dict*) – The data from a previous session we are continuing.

classmethod **_calculate** ()

Calculate the percentage of each status.

count ()

Count the number of domain for each status.

log ()

Print on screen and on file the percentages for each status.

12.20 Prints

12.20.1 Problematic

How can we print information on the screen and on file in a table format?

12.20.2 Documentation

class PyFunceble.prints.**Prints** (*to_print, template, output_file=None, only_on_file=False*)

Print data on screen and into a file if needed. Template Possibilities: Percentage, Less, HTTP and any status you want.

Parameters

- **to_print** – The list of data to print.
- **template** (*str*) – The template to use.

Note: Available templates:

- Percentage
 - Less
 - HTTP
 - any of the official status.
-

- **output_file** (*str*) – The path to the file to write.
- **only_on_file** (*bool*) – Tell us if we only have to print on file and not on screen.

__before_header ()

Print informations about PyFuncible and the date of generation of a file into a given path, if doesn't exist.

__colorify (*data*)

Return colored string.

Parameters **data** (*str*) – The string to colorify.

Returns A colored string.

Return type str

__data_constructor (*size*)

Construct the table of data according to given size.

Parameters **size** (*list*) – The maximal length of each string in the table.

Returns A dict with all information about the data and how to which what maximal size to print it.

Return type OrderedDict

Raises

Exception If the data and the size does not have the same length.

classmethod **__header_constructor** (*data_to_print*, *header_separator='-'*, *column_separator=' '*)

Construct header of the table according to template.

Parameters

- **data_to_print** (*list*) – The list of data to print into the header of the table.
- **header_separator** (*str*) – The separator to use between the table header and our data.
- **column_separator** (*str*) – The separator to use between each columns.

Returns The data to print in a list format.

Return type list

__json_print ()

Management of the json template.

classmethod **__size_from_header** (*header*)

Get the size of each columns from the header.

Parameters **header** (*dict*) – The header template we have to get the size from.

Returns The maximal size of the each data to print.

Return type list

data()

Management and input of data to the table.

Raises

Exception When `self.data_to_print` is not a list.

header (*do_not_print=False*)

Management and creation of templates of header. Please consider as “header” the title of each columns.

Parameters `do_not_print` (*bool*) – Tell us if we have to print the header or not.

12.21 Production

12.21.1 Problematic

How can we efficiently prepare the repository for push/production?

12.21.2 Documentation

class `PyFunceble.production.Production` (*extern=False*)

Manage and provide the production preparation logic.

Parameters `extern` (*bool*) – Tell us if we do not have to execute the logic automatically. This allow method to be called.

classmethod `is_dev_version()`

Check if the current branch is *dev*.

classmethod `is_master_version()`

Check if the current branch is *master*.

12.22 Public Suffix

12.22.1 Problematic

How can we get the list of all possible or at least most used domain suffix?

12.22.2 Documentation

class `PyFunceble.publicsuffix.PublicSuffix`

Let us interact with the public suffix database.

classmethod `_data()`

Get the database from the public suffix repository.

_extensions (*line*)

Extract the extension from the given line.

Parameters `line` (*str*) – The line from the official public suffix repository.

load()

Load the public suffix database into the system.

update()
Update of the content of the `public-suffix.json`.

12.23 Referer

12.23.1 Problematic

How can we efficiently get the whois server to call for whois record request?

12.23.2 Documentation

class `PyFuncible.referer.Referer`

Get the WHOIS server (referer) of the current domain extension according to the IANA database.

get()
Return the referer aka the WHOIS server of the current domain extension.

12.24 Sort

12.24.1 Problematic

How can we format the list to test (and the outputted information) in a format other than the alphabetical format?

12.24.2 Documentation

class `PyFuncible.sort.Sort`

Provide some sorting presets which we can parse to `PyFuncible.helpers.List().custom_format()`.

classmethod `hierarchical(element)`

The idea behind this method is to sort a list of domain hierarchically.

Parameters `element` (*str*) – The element we are currently reading.

Returns The formatted element.

Return type `str`

Note: For a domain like `aaa.bbb.ccc.tdl`.

A normal sorting is done in the following order:

1. `aaa`
2. `bbb`
3. `ccc`
4. `tdl`

This method allow the sorting to be done in the following order:

1. `tdl`
2. `ccc`

3. bbb

4. aaa

classmethod standard (*element*)

Implement the standard and alphabetical sorting.

Parameters **element** (*str*) – The element we are currently reading.

Returns The formatted element.

Return type *str*

12.25 Status

12.25.1 Problematic

How can we efficiently manage the statuses in function of the test type?

12.25.2 Documentation

Normal testing

class `PyFunceble.status.Status`

Handle the research of domain status in case we don't use WHOIS or in case that WHOIS record is not readable nor exploitable.

Parameters **matched_result** (*str*) – The previously caught status.

class `ExtraRules`

Manage some extra rules.

`__ExtraRules__blogspot` ()

Handle the blogspot SPECIAL case.

Returns (*new status, new source*) or `None` if there is any change to apply.

Return type `tuple|None`

`__ExtraRules__handle_http_code` (*previous_state*)

Handle the HTTP Code status escalation.

Parameters **previous_state** (*str*) – The previously caught status.

Returns (*new status, new source*) or `None` if there is any change to apply.

Return type `tuple|None`

`__ExtraRules__handle_ipv4_range` ()

Handle the IP range status escalation.

Returns (*new status, new source*) or `None` if there is any change to apply.

Return type `tuple|None`

`__ExtraRules__handle_potentially_inactive` (*previous_state*)

Handle the potentially inactive case.

Parameters **previous_state** (*str*) – The previously caught status.

Returns (*new status, new source*) or `None` if there is any change to apply.

Return type `tuple|None`

`__ExtraRules__handle_potentially_up` ()

Handle the potentially up case.

Returns (new status, new source) or None if there is any change to apply.

Return type tuple|None

classmethod `__ExtraRules__http_status_code_up()`

Set what we return for the HTTP Code status escalation.

Returns (new status, new source)

Return type tuple

classmethod `__ExtraRules__special_down()`

Set what we return for the SPECIAL status de-escalation.

Returns (new status, new source)

Return type tuple

classmethod `__ExtraRules__special_up()`

Set what we return for the SPECIAL status escalation.

Returns (new status, new source)

Return type tuple

__ExtraRules__wordpress_dot_com()

Handle the wordpress.com SPECIAL case.

Returns (new status, new source) or None if there is any change to apply.

Return type tuple|None

handle (*previous_state*, *previous_source*)

Globally handle the case of the currently tested domain.

classmethod `get()`

Get the status while testing for an IP or domain.

Note: We consider that the domain or IP we are currently testing is into `PyFunceble.INTERN["to_test"]`.

classmethod `handle(status, invalid_source='IANA')`

Handle the lack of WHOIS and expiration date. `:smile_cat:`

Parameters

- **matched_status** – The status that we have to handle.
- **invalid_source** (*str*) – The source to set when we handle INVALID element.

Returns The strus of the domain after generating the files desired by the user.

Return type str

URL testing

class `PyFunceble.status.URLStatus(catched_status)`

Generate everything around the caught status when testing for URL.

Parameters `catched_status` (*str*) – THE caught status.

handle ()

Handle the backend of the given status.

Syntax checking

```
class PyFunceble.status.SyntaxStatus (caught_status)
    Generate everything around the caught status when testing for Syntax.

    Parameters caught_status (str) – The caught status.

handle ()
    Handle the backend of the given status.
```

12.26 Syntax Checking

12.26.1 Problematic

How can we check for syntax directly from the CLI?

12.26.2 Documentation

```
class PyFunceble.syntax.Syntax
    Manage everything around the Syntax testing.

    classmethod get ()
        Execute the logic behind the Syntax handling.

        Returns The syntax status.

        Return type str
```

12.27 URL Testing

12.27.1 Problematic

How can we test full URL?

12.27.2 Documentation

```
class PyFunceble.url.URL
    Manage everything around the URL testing.

    classmethod get ()
        Execute the logic behind the URL handling.

        Returns The status of the URL.

        Return type str
```

CHAPTER 13

Known issues

This is the list of issues which are or will not be fixed (yet?).

- Under Travis CI the coloration may not be shown.

Frequently Asked Questions

Note: Don't want to leave your machine running PyFuncible for days?

Request your list to be tested at [Dead-Hosts!](#)

14.1 How to speed up a bit the process/test?

For now, we only invite you to disable the usage/request of the WHOIS record.

This action will let the script speed up because we only use the equivalent of `nslookup` and the HTTP status code to determine the status.

Warning: We use/request the WHOIS record in order to avoid a specific false positive case. Indeed, if we disable the usage/request all domains which are still registered but not assigned to a specific IP will be flagged as INACTIVE.

It's not a problem if you keep/use the database system because the domain will be retested over time. But please keep in mind that without the database system we do not guarantee an accurate result.

I'm really glad you're reading this because we need contributions to make this tool one of the best tool on the Internet!

15.1 Submitting changes

Before anything, please keep in mind the following. If one or more of those conditions are not filled. Your Pull Request to PyFunceble may not be merged.

The `master` branch is used only for releasing a new or stable version of the code. That's why we require that all contributions/modifications must be done under the `dev` or a new branch.

In order to gain some time and also understand what you are working on, your pull requests submission as long as your commit descriptions should be clear and complete as much as possible. We do an exception to commit with minor changed but big changes should have a complete description. Please ensure to use the following method when committing a big change.

15.1.1 Steps before commit

Note: The following do not apply if you do not touch the `PyFunceble` nor the `tests` directory.

```
$ # We format our code.
$ black PyFunceble && black tests/*.py
$ # We lint our code. Please make sure to fix all reported issues.
$ pylint PyFunceble && pylint tests/*.py
$ # We check the tests coverage. Please ensure that at least 95% of the code is_
->covered.
$ coverage run setup.py test && coverage report -m
$ # Prepare our files, :code:`version.yaml` and code for pushing.
$ PyFunceble --production
```

15.1.2 The commit

```
$ # There paragraph is optional if your changes/commits are obvious.
$ git commit -S -m "A summary of the commit" -m "A paragraph
> or a sentence explaining what changed, why and its impact."
```

All your commits should be signed with **PGP**. (More information can be found on [GitHub documentation](#))

Please note the usage of `-S` into the commit command which means that we sign the commit. The usage of `PyFunceble --production update version.yaml` and `directory_structure_production.json` automatically.

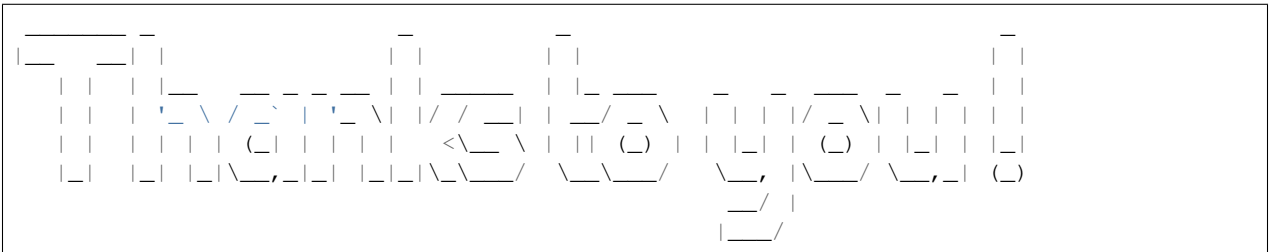
15.2 Coding conventions

- We make sure that a method, a function, and a class **have doctstrings**.
- One line should not exceed 79 characters for docstring and 100 characters for long declaration/assignment. - Exception granted for regular expressions or long string assignment.
- We use [Black](#), *The uncompromising Python code formatter*, to format our code.
- Our code should pass `pylint PyFunceble && pylint tests/*.py` with at least a score of 10.00/10.00
- We do not forget to follow the steps before any commits.

CHAPTER 16

Contributors

Thanks to those awesome peoples for their awesome and crazy idea(s), contribution(s) and or issue report which made or make [PyFuncible](#) a better tool.



- dnmTX - @dnmTX
- hawkeye116477 - @hawkeye116477
- Imre Kristoffer Eilertsen - @DandelionSprout
- jawz101 - @jawz101
- Mitchell Krog - @mitchellkrogza
- Odysseus - @Odysseus
- Reza Rizqullah - @ybreza
- ScriptTiger - @ScriptTiger
- speedmann - @speedmann
- WaLLy3K - @WaLLy3K
- xxcriticxx - @xxcriticxx

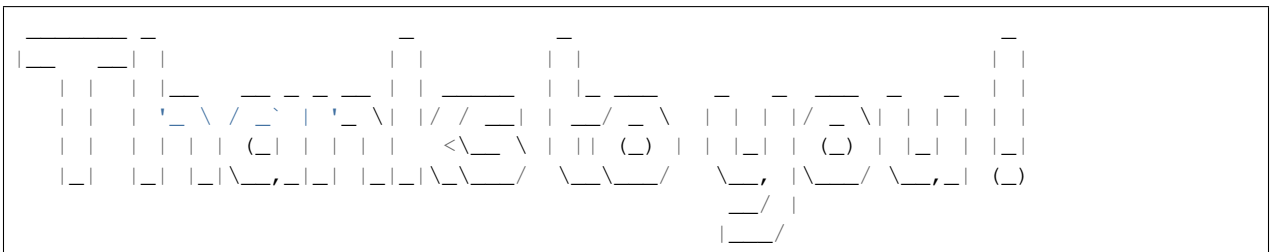
CHAPTER 17

Special Thanks

Thanks to those awesome organization(s), tools and or people(s) for

- Their awesome documentation
- Their awesome repository
- Their awesome tool/software/source code
- Their breaking reports
- Their contributions
- Their current work/purpose
- Their promotion of Py-Funceleble
- Their support
- Their testings reports

which helped and/or still help me build, test and or make [PyFunceleble](#) a better tool.



- Adam Warner - [@PromoFaux](#)
- Adblock Plus - [@adblockplus](#)
- Mitchell Krog - [@mitchellkrogza](#)
- Mohammad Fares - [@faressoft](#)
- Pi-Hole - [@pi-hole](#)

- Public Suffix List - @publicsuffix
- Reza Rizqullah - @ybreza
- ScriptTiger - @ScriptTiger
- SMed79 - @SMed79
- IANA - ICANN
- yEd Graph Editor - yWorks

CHAPTER 18

Supporting the project

PyFuncible, Dead-Hosts, and all other analog projects are powered by free time and a lot of coffee!

This project helps you and/or you like it?

 Buy Me a Coffee

CHAPTER 19

They use PyFunceble!

Just a list of tool/repository/code/infrastructure which uses or used PyFunceble!

- .pl-host-file
- add.2o7Net
- add.Dead
- add.Risk
- add.Spam
- adfilt
- Badd-Boyz-Hosts
- Badger Sett
- blacklist
- CoinBlockerLists
- cookies_filters
- Facebook Zero Hosts Block
- finnish-easylist-addition
- Frellwits-filter-lists
- Google AdService and ID.Google tracking hosts
- KADhosts
- MobileAdTrackers
- Phishing-URL-Testing-Database-of-Link-Statuses
- Phishing.Database
- polish-adblock-filters
- polish-pihole-filters

- [Stop.Google.Analytics.Ghost.Spam.HOWTO](#)
- [The-Big-List-of-Hacked-Malware-Web-Sites](#)
- [Top-Attacking-IP-Addresses-Against-Wordpress-Sites](#)
- [Ultimate.Hosts.Blacklist](#)
- [UncheckyAds](#)

CHAPTER 20

License

MIT License

Copyright (c) 2017–2019 Nissar Chababy

Permission **is** hereby granted, free of charge, to **any** person obtaining a copy of this software **and** associated documentation files (the "**Software**"), to deal **in** the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, **and/or** sell copies of the Software, **and** to permit persons to whom the Software **is** furnished to do so, subject to the following conditions:

The above copyright notice **and** this permission notice shall be included **in all** copies **or** substantial portions of the Software.

THE SOFTWARE IS PROVIDED "**AS IS**", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

21.1 Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation.

21.2 Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

21.3 Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

21.4 Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

21.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at contact@funilrys.com. The project team will review and investigate all complaints, and will respond in a way that it deems appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

21.6 Attribution

This Code of Conduct is adapted from the [Contributor Covenant](http://contributor-covenant.org/version/1/4), version 1.4, available at <http://contributor-covenant.org/version/1/4>.

CHAPTER 22

Discovered any issue?

Want to make it public? Then fill a [new issue](#).

Want to talk with [@funilrys](#) about it? Then stay in touch with [Fun Ilrys on Keybase](#) (preferred) or send an email to contactTATATAfunilrysTODTOTODcom.

Don't want to fill an issue on GitHub? You can still inform us about it on our [issues team](#) on Keybase.

CHAPTER 23

Need help?

Do you need help with the usage or application of PyFuncible? We can help!

Want to talk with [@funilrys](#) about it? Then stay in touch with [Fun Ilrys on Keybase](#) (preferred) or send an email to contactTATATAfunilrysTODTOTODcom.

Otherwise, let's talk about it on our [support team](#) or our [main team](#) on Keybase!

CHAPTER 24

Indices and tables

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

p

`PyFunceble`, [77](#)

`PyFunceble.helpers`, [81](#)

Symbols

- `_ExtraRules__blogspot()` (*PyFunceble.status.Status.ExtraRules method*), 112
- `_ExtraRules__handle_http_code()` (*PyFunceble.status.Status.ExtraRules method*), 112
- `_ExtraRules__handle_ipv4_range()` (*PyFunceble.status.Status.ExtraRules method*), 112
- `_ExtraRules__handle_potentially_inactive()` (*PyFunceble.status.Status.ExtraRules method*), 112
- `_ExtraRules__handle_potentially_up()` (*PyFunceble.status.Status.ExtraRules method*), 112
- `_ExtraRules__http_status_code_up()` (*PyFunceble.status.Status.ExtraRules class method*), 113
- `_ExtraRules__special_down()` (*PyFunceble.status.Status.ExtraRules class method*), 113
- `_ExtraRules__special_up()` (*PyFunceble.status.Status.ExtraRules class method*), 113
- `_ExtraRules__wordpress_dot_com()` (*PyFunceble.status.Status.ExtraRules method*), 113
- `_access()` (*PyFunceble.http_code.HTTPCode method*), 105
- `_add()` (*PyFunceble.mining.Mining method*), 107
- `_add_to_test()` (*PyFunceble.database.Inactive method*), 99
- `_analytic_host_file_directory()` (*PyFunceble.generate.Generate method*), 104
- `_authorization()` (*PyFunceble.database.whois class method*), 100
- `_authorization()` (*PyFunceble.execution_time.ExecutionTime class method*), 102
- `_backup()` (*PyFunceble.database.Inactive method*), 99
- `_backup()` (*PyFunceble.database.whois method*), 100
- `_backup()` (*PyFunceble.mining.Mining method*), 107
- `_before_header()` (*PyFunceble.prints.Prints method*), 109
- `_calculate()` (*PyFunceble.execution_time.ExecutionTime class method*), 102
- `_calculate()` (*PyFunceble.percentage.Percentage class method*), 108
- `_cases_management()` (*PyFunceble.expiration_date.ExpirationDate method*), 103
- `_colorify()` (*PyFunceble.prints.Prints method*), 109
- `_convert_1_to_2_digits()` (*PyFunceble.expiration_date.ExpirationDate class method*), 103
- `_convert_or_shorten_month()` (*PyFunceble.expiration_date.ExpirationDate class method*), 103
- `_create_directory()` (*PyFunceble.directory_structure.DirectoryStructure class method*), 101
- `_data()` (*PyFunceble.publicsuffix.PublicSuffix class method*), 110
- `_data_constructor()` (*PyFunceble.prints.Prints method*), 109
- `_decode_output()` (*PyFunceble.helpers.Command method*), 82, 87
- `_do_not_produce_file()` (*PyFunceble.generate.Generate method*), 104
- `_entry_management()` (*PyFunceble.core.Core method*), 97
- `_entry_management_url()` (*PyFunceble.core.Core method*), 97
- `_entry_management_url_download()` (*PyFunceble.core.Core method*), 97
- `_extensions()` (*PyFunceble.iana.IANA method*), 105
- `_extensions()` (*PyFunceble*), 105

ble.publicsuffix.PublicSuffix method), 110
 __extract() (PyFunceble.ble.expiration_date.ExpirationDate method), 103
 __extract_base() (PyFunceble.adblock.AdBlock method), 91
 __extract_domain_from_file() (PyFunceble.core.Core class method), 97
 __file_decision() (PyFunceble.core.Core method), 97
 __file_list_to_test_filtering() (PyFunceble.core.Core method), 97
 __format() (PyFunceble.ble.expiration_date.ExpirationDate method), 103
 __format_decoded() (PyFunceble.adblock.AdBlock method), 91
 __format_domain() (PyFunceble.core.Core class method), 97
 __get_content() (PyFunceble.logs.Logs class method), 106
 __get_structure() (PyFunceble.ble.directory_structure.DirectoryStructure method), 101
 __handle_non_existant_index() (PyFunceble.ble.generate.Generate class method), 104
 __handle_options() (PyFunceble.adblock.AdBlock method), 91
 __hash_data() (PyFunceble.helpers.Hash method), 85, 91
 __hash_file() (PyFunceble.helpers.Hash method), 85, 91
 __header_constructor() (PyFunceble.ble.prints.Prints class method), 109
 __install_directory_structure_file() (PyFunceble.config.Load class method), 95
 __install_iana_config() (PyFunceble.config.Load class method), 95
 __install_production_config() (PyFunceble.config.Load method), 95
 __install_psl_config() (PyFunceble.config.Load class method), 95
 __is_to_ignore() (PyFunceble.adblock.AdBlock class method), 92
 __json_print() (PyFunceble.prints.Prints method), 109
 __load_config_file() (PyFunceble.config.Load method), 95
 __merge() (PyFunceble.database.Inactive method), 99
 __print_header() (PyFunceble.core.Core class method), 97
 __prints_status_file() (PyFunceble.ble.generate.Generate method), 104
 __prints_status_screen() (PyFunceble.ble.generate.Generate method), 104
 __referer() (PyFunceble.iana.IANA method), 105
 __reformat_historical_formatting_error() (PyFunceble.database.Inactive method), 99
 __remove_ignored() (PyFunceble.adblock.AdBlock method), 92
 __restore_replace() (PyFunceble.ble.directory_structure.DirectoryStructure method), 101
 __retrieve() (PyFunceble.database.Inactive method), 100
 __retrieve() (PyFunceble.database.whois.whois method), 100
 __retrieve() (PyFunceble.mining.Mining method), 107
 __save() (PyFunceble.execution_time.ExecutionTime method), 102
 __set_path_to_configs() (PyFunceble.config.Load class method), 95
 __size_from_header() (PyFunceble.prints.Prints class method), 109
 __starting_time() (PyFunceble.execution_time.ExecutionTime class method), 102
 __stopping_time() (PyFunceble.execution_time.ExecutionTime class method), 102
 __timestamp() (PyFunceble.database.Inactive method), 100
 __travis() (PyFunceble.auto_save.AutoSave method), 93
 __update_structure_from_config() (PyFunceble.ble.directory_structure.DirectoryStructure method), 101
 __write_content() (PyFunceble.logs.Logs class method), 106

A

AdBlock (class in PyFunceble.adblock), 91
 add() (PyFunceble.database.Inactive method), 100
 add() (PyFunceble.database.whois.whois method), 100
 analytic_file() (PyFunceble.generate.Generate method), 104
 AutoContinue (class in PyFunceble.auto_continue), 92
 AutoSave (class in PyFunceble.auto_save), 93

B

backup() (PyFunceble.auto_continue.AutoContinue method), 92
 backup() (PyFunceble.ble.directory_structure.DirectoryStructure method), 101
 bypass() (PyFunceble.core.Core class method), 97

C

Check (class in *PyFunceble.check*), 93
 check_versions() (*PyFunceble.config.Version* class method), 95
 check_versions_literally() (*PyFunceble.config.Version* class method), 95
 Clean (class in *PyFunceble.clean*), 94
 colorify_logo() (*PyFunceble.core.Core* class method), 97
 Command (class in *PyFunceble.helpers*), 82, 87
 compare() (*PyFunceble.config.Version* method), 96
 content() (*PyFunceble.database.Inactive* class method), 100
 copy() (*PyFunceble.helpers.File* method), 84, 90
 Core (class in *PyFunceble.core*), 96
 count() (*PyFunceble.percentage.Percentage* method), 108
 custom_format() (*PyFunceble.helpers.List* method), 85, 89

D

data() (*PyFunceble.prints.Prints* method), 109
 decode() (*PyFunceble.adblock.Adblock* method), 92
 delete() (*PyFunceble.helpers.File* method), 84, 90
 delete_unneeded() (*PyFunceble.directory_structure.DirectoryStructure* method), 101
 Dict (class in *PyFunceble.helpers*), 82, 88
 Directory (class in *PyFunceble.helpers*), 83, 90
 DirectoryStructure (class in *PyFunceble.directory_structure*), 101
 domain() (*PyFunceble.core.Core* method), 97
 Download (class in *PyFunceble.helpers*), 84, 86

E

execute() (*PyFunceble.helpers.Command* method), 82, 87
 ExecutionTime (class in *PyFunceble.execution_time*), 102
 expiration_date() (*PyFunceble.logs.Logs* method), 106
 ExpirationDate (class in *PyFunceble.expiration_date*), 103

F

File (class in *PyFunceble.helpers*), 84, 90
 file() (*PyFunceble.core.Core* method), 98
 file_url() (*PyFunceble.core.Core* method), 98
 fix_path() (*PyFunceble.helpers.Directory* method), 84, 90
 format() (*PyFunceble.helpers.List* method), 85, 89
 format_execution_time() (*PyFunceble.execution_time.ExecutionTime* method), 102

from_json() (*PyFunceble.helpers.Dict* class method), 82, 88
 from_yaml() (*PyFunceble.helpers.Dict* class method), 83, 88

G

Generate (class in *PyFunceble.generate*), 104
 get() (*PyFunceble.expiration_date.ExpirationDate* method), 104
 get() (*PyFunceble.helpers.Hash* method), 85, 91
 get() (*PyFunceble.http_code.HTTPCode* method), 105
 get() (*PyFunceble.referer.Referer* method), 111
 get() (*PyFunceble.status.Status* class method), 113
 get() (*PyFunceble.syntax.Syntax* class method), 114
 get() (*PyFunceble.url.URL* class method), 114
 get_expiration_date() (*PyFunceble.database.whois* method), 100

H

handle() (*PyFunceble.status.Status* class method), 113
 handle() (*PyFunceble.status.Status.ExtraRules* method), 113
 handle() (*PyFunceble.status.SyntaxStatus* method), 114
 handle() (*PyFunceble.status.URLStatus* method), 113
 Hash (class in *PyFunceble.helpers*), 84, 90
 header() (*PyFunceble.prints.Prints* method), 110
 hierarchical() (*PyFunceble.sort.Sort* class method), 111
 HTTPCode (class in *PyFunceble.http_code*), 105

I

IANA (class in *PyFunceble.iana*), 105
 Inactive (class in *PyFunceble.database*), 99
 info_files() (*PyFunceble.generate.Generate* method), 104
 ipv4_syntax_check() (in module *PyFunceble*), 78
 is_cloned() (*PyFunceble.config.Version* class method), 96
 is_dev_version() (*PyFunceble.production.Production* class method), 110
 is_domain_valid() (*PyFunceble.check.Check* method), 93
 is_in_database() (*PyFunceble.database.whois* method), 100
 is_ip_range() (*PyFunceble.check.Check* method), 93
 is_ip_valid() (*PyFunceble.check.Check* method), 93
 is_ipv4_range() (in module *PyFunceble*), 78

- `is_master_version()` (*PyFunceble.production.Production class method*), 110
- `is_present()` (*PyFunceble.database.Inactive class method*), 100
- `is_subdomain()` (*in module PyFunceble*), 78
- `is_subdomain()` (*PyFunceble.check.Check method*), 94
- `is_time_older()` (*PyFunceble.database.whois method*), 100
- `is_url_valid()` (*PyFunceble.check.Check method*), 94
- ## L
- `List` (*class in PyFunceble.helpers*), 85, 89
- `list_of_mined()` (*PyFunceble.mining.Mining class method*), 107
- `Load` (*class in PyFunceble.config*), 95
- `load()` (*PyFunceble.iana.IANA method*), 105
- `load()` (*PyFunceble.publicsuffix.PublicSuffix method*), 110
- `load_config()` (*in module PyFunceble*), 78
- `log()` (*PyFunceble.percentage.Percentage method*), 108
- `Logs` (*class in PyFunceble.logs*), 106
- `Lookup` (*class in PyFunceble.lookup*), 107
- ## M
- `match()` (*PyFunceble.helpers.Regex method*), 86, 88
- `matching_list()` (*PyFunceble.helpers.Regex method*), 86, 88
- `merge()` (*PyFunceble.helpers.Dict method*), 83, 88
- `merge()` (*PyFunceble.helpers.List method*), 85, 89
- `mine()` (*PyFunceble.mining.Mining method*), 107
- `Mining` (*class in PyFunceble.mining*), 107
- ## N
- `not_matching_list()` (*PyFunceble.helpers.Regex method*), 86, 88
- `nslookup()` (*PyFunceble.lookup.Lookup class method*), 107
- ## P
- `Percentage` (*class in PyFunceble.percentage*), 108
- `Prints` (*class in PyFunceble.prints*), 108
- `process()` (*PyFunceble.mining.Mining method*), 108
- `Production` (*class in PyFunceble.production*), 110
- `PublicSuffix` (*class in PyFunceble.publicsuffix*), 110
- `PyFunceble` (*module*), 77
- `PyFunceble.helpers` (*module*), 81
- ## R
- `read()` (*PyFunceble.helpers.File method*), 84, 90
- `Referer` (*class in PyFunceble.referer*), 111
- `referer_not_found()` (*PyFunceble.logs.Logs method*), 106
- `Regex` (*class in PyFunceble.helpers*), 86, 87
- `remove()` (*PyFunceble.database.Inactive method*), 100
- `remove()` (*PyFunceble.mining.Mining method*), 108
- `remove_key()` (*PyFunceble.helpers.Dict method*), 83, 88
- `rename_key()` (*PyFunceble.helpers.Dict method*), 83, 89
- `replace()` (*PyFunceble.helpers.Regex method*), 86, 88
- `reset_counters()` (*PyFunceble.core.Core class method*), 98
- `restore()` (*PyFunceble.auto_continue.AutoContinue method*), 92
- `restore()` (*PyFunceble.directory_structure.DirectoryStructure method*), 101
- `right_url_from_version()` (*PyFunceble.config.Version class method*), 96
- `run()` (*PyFunceble.helpers.Command method*), 82, 87
- ## S
- `Sort` (*class in PyFunceble.sort*), 111
- `split_versions()` (*PyFunceble.config.Version class method*), 96
- `standard()` (*PyFunceble.sort.Sort class method*), 112
- `Status` (*class in PyFunceble.status*), 112
- `Status.ExtraRules` (*class in PyFunceble.status*), 112
- `status_file()` (*PyFunceble.generate.Generate method*), 104
- `switch()` (*PyFunceble.core.Core class method*), 98
- `Syntax` (*class in PyFunceble.syntax*), 114
- `syntax_check()` (*in module PyFunceble*), 79
- `SyntaxStatus` (*class in PyFunceble.status*), 114
- ## T
- `test()` (*in module PyFunceble*), 79
- `test()` (*PyFunceble.core.Core method*), 98
- `test_with_complete_information()` (*PyFunceble.core.Core method*), 98
- `text()` (*PyFunceble.helpers.Download method*), 84, 86
- `to_json()` (*PyFunceble.helpers.Dict method*), 83, 89
- `to_test()` (*PyFunceble.database.Inactive method*), 100
- `to_yaml()` (*PyFunceble.helpers.Dict method*), 83, 89
- `travis_permissions()` (*PyFunceble.auto_save.AutoSave class method*), 93
- ## U
- `unified_file()` (*PyFunceble.generate.Generate method*), 105
- `update()` (*PyFunceble.iana.IANA method*), 106

update() (*PyFunceble.publicsuffix.PublicSuffix method*), 110

URL (*class in PyFunceble.url*), 114

url() (*PyFunceble.core.Core method*), 99

url_syntax_check() (*in module PyFunceble*), 79

url_test() (*in module PyFunceble*), 80

URLStatus (*class in PyFunceble.status*), 113

V

Version (*class in PyFunceble.config*), 95

W

Whois (*class in PyFunceble.database*), 100

whois() (*PyFunceble.logs.Logs method*), 106

whois() (*PyFunceble.lookup.Lookup class method*), 107

write() (*PyFunceble.helpers.File method*), 84, 90