
qiuwenbot Documentation

Jinzhe Zeng

Apr 17, 2024

CONTENTS

1	Features /	3
2	Disclaimer /	5
3	Table of Contents	7
3.1	Installation	7
3.2	Usage	7
3.3	Command line interface	7
3.4	Input Parameters	8
3.5	API documentation	10
4	Indices and tables	33
	Python Module Index	35
	Index	37

Qiuwen Baike bot. /

FEATURES /

- Comprehensive Cleanup /
 - Cleaning up illegal references /
 - Cleaning up unsupported HTML tags / HTML
 - Cleaning up expired templates /
 - Removing illegal flags of the Taiwan authorities /
 - Removing illegal era names of the Taiwan authorities /
 - Correcting terms related to Taiwan /
 - Correcting terms related to Hong Kong /
 - Correcting terms related to politics /
 - Correcting terms related to “Cultural Revolution” / “”
- Marking redundant entries caused by differences between Simplified and Traditional Chinese /

DISCLAIMER /

Qiuwen Baike® is a trademark or registered trademark of the operator of Qiuwen Baike website or its affiliated entities. This project uses the trademark legitimately based on Article 59 of the “Trademark Law of the People’s Republic of China”. When using it in the modified version of this project, one should observe the “Trademark Law”. / ®

TABLE OF CONTENTS

3.1 Installation

To install qiuwenbot, simply use pip:

```
pip install qiuwenbot
```

3.2 Usage

Firstly, one needs to prepare a JSON file that describes the job.

```
1 {  
2   "user": "Njzjzbot",  
3   "task": "filter",  
4   "pages": {  
5     "type": "all"  
6   }  
7 }
```

The detailed parameters can be found in *Input Parameters*. The password can be set by `QIUWENBOT_PASSWORD` environment variable. It is recommended using the bot password.

Then, one can submit the job using the following command:

```
qiuwenbot submit filter_all.json
```

See *Command line interface* for more information.

3.3 Command line interface

```
usage: qiuwenbot [-h] {submit,gui} ...
```

3.3.1 Sub-commands

submit

Submit a task

```
qiuwenbot submit [-h] CONFIG
```

Positional Arguments

CONFIG	Path to the config file
---------------	-------------------------

gui

Serve DP-GUI.

```
qiuwenbot gui [-h] [-p PORT] [--bind_all]
```

Named Arguments

-p, --port	The port to serve DP-GUI on. Default: 6042
--bind_all	Serve on all public interfaces. This will expose your DP-GUI instance to the network on both IPv4 and IPv6 (where available). Default: False

3.4 Input Parameters

Note: One can load, modify, and export the input file by using our effective web-based tool DP-GUI hosted using the *command line interface* `qiuwenbot gui`. All parameters below can be set in DP-GUI. By clicking “SAVE JSON”, one can download the input file.

user:

type: str
argument path: user
Username.

pages:

type: dict
argument path: pages
Configurations of scanned pages.
Depending on the value of *type*, different sub args are accepted.

type:

type: str (flag key)
 argument path: pages/type
 possible choices: *all*, *new*, *link*, *page*
 Method to scan pages.

When *type* is set to *all*:

Scan all pages in alphabetical order.

namespace:

type: int, optional, default: 0
 argument path: pages[all]/namespace
 Namespace(s) of the pages.

restart:

type: bool, optional, default: False
 argument path: pages[all]/restart
 Restart from the last page in the log.

When *type* is set to *new*:

Scan new pages.

namespace:

type: list | int, optional, default: 0
 argument path: pages[new]/namespace
 Namespace(s) of the pages.

start:

type: str | NoneType, optional, default: None
 argument path: pages[new]/start
 Start time in ISO format.

end:

type: str | NoneType, optional, default: None
 argument path: pages[new]/end
 End time in ISO format.

When *type* is set to *link* (or its alias *template*):

Scan pages that link to a page or include a template.

name:

type: str
 argument path: pages[link]/name
 Name of the page or template.

namespace:

type: list | int | NoneType, optional, default: None
 argument path: pages[link]/namespace
 Namespace(s) of the pages.

When `type` is set to `page`:

Scan a single page.

name:

type: `str`

argument path: `pages[page]/name`

Name of the page or template.

task:

type: `str`

argument path: `task`

Task to submit.

3.5 API documentation

3.5.1 qiuwenbot package

Qiuwen bot.

Subpackages

qiuwenbot.entrypoints package

Scripts.

Submodules

qiuwenbot.entrypoints.gui module

DP-GUI entrypoint.

`qiuwenbot.entrypoints.gui.start_dpgui` (*args: Namespace*)

Host DP-GUI server.

Parameters

args

[`argparse.Namespace`] Arguments from `argparse`.

Raises

ModuleNotFoundError

The `dpgui` package is not installed

qiuwenbot.entrypoints.submit module

qiuwenbot.entrypoints.submit.**submit**(args: *Namespace*)

Submit a task.

qiuwenbot.filter package

Filter texts.

Submodules

qiuwenbot.filter.clean_refs module

class qiuwenbot.filter.clean_refs.CleanRefsFilter

Bases: *Filter*

Filter to clean references.

Attributes

log

Log of the filter.

Methods

<i>filter</i> (text)	Filter text.
----------------------	--------------

filter(text: *str*) → *str*

Filter text.

Parameters

text

[*str*] Text to filter.

Returns

str

Filtered text.

property log: *str*

Log of the filter.

Returns

str

Log of the filter.

qiuwenbot.filter.common module

`qiuwenbot.filter.common.get_comment(comment: str) → str`

Get comment inserted into wikitext.

qiuwenbot.filter.deprecated_tags module

class `qiuwenbot.filter.deprecated_tags.RemoveMapframeFilter`

Bases: *RemoveTagFilter*

Filter to remove mapframe tag.

Attributes

log

Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

class `qiuwenbot.filter.deprecated_tags.RemoveScoreFilter`

Bases: *RemoveTagFilter*

Filter to remove score tag.

Attributes

log

Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

class `qiuwenbot.filter.deprecated_tags.RemoveTagFilter(tag: str)`

Bases: *TextReplaceFilter*

Filter to remove a certain tag.

Parameters

tag

[str] Tag name to remove.

Attributes

log

Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

property log: `str`

Log of the filter.

Returns

`str`

Log of the filter.

class `qiuwenbot.filter.deprecated_tags.RemoveTimelineFilter`

Bases: *RemoveTagFilter*

Filter to remove timeline tag.

Attributes

log

Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

qiuwenbot.filter.expired_templates module

class `qiuwenbot.filter.expired_templates.RemoveExpiredCurrentFilter`

Bases: *RemoveExpiredTemplateFilter*

Filter to remove {{current}}.

Attributes

log

Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

class `qiuwenbot.filter.expired_templates.RemoveExpiredDeadFilter`

Bases: *RemoveExpiredTemplateFilter*

Filter to remove {{}}.

Attributes

log

Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

class `qiuwenbot.filter.expired_templates.RemoveExpiredDeadFilter2`

Bases: *RemoveExpiredTemplateFilter*

Filter to remove `{{}}`.

Attributes

log

Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

class `qiuwenbot.filter.expired_templates.RemoveExpiredDeadFilter3`

Bases: *RemoveExpiredTemplateFilter*

Filter to remove `{{recent death}}`.

Attributes

log

Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

class `qiuwenbot.filter.expired_templates.RemoveExpiredTemplateFilter(template: str)`

Bases: *Filter*

Filter to remove a certain tag.

Parameters

tag

[str] Tag name to remove.

Attributes

log

Log of the filter.

Methods

<i>filter</i> (text)	Filter text.
----------------------	--------------

filter(*text: str*) → str

Filter text.

Parameters

text
[str] Text to filter.

Returns

str
Filtered text.

property log: **str**

Log of the filter.

Returns

str
Log of the filter.

qiuwenbot.filter.filter module

class qiuwenbot.filter.filter.**Filter**

Bases: `object`

Filter texts.

Attributes

log
Log of the filter.

Methods

<i>filter</i> (text)	Filter text.
----------------------	--------------

abstract filter(*text: str*) → str

Filter text.

Parameters

text
[str] Text to filter.

Returns

str
Filtered text.

property log: `str | None`

Log of the filter.

Returns

`str`

Log of the filter.

class `qiuwenbot.filter.filter.FilterChain(filters: List[Filter])`

Bases: `Filter`

Filter chain.

Parameters

filters

[list of Filter] Filters to apply.

Attributes

log

Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

filter(*text: str*) → `str`

Filter text.

Parameters

text

[str] Text to filter.

Returns

`str`

Filtered text.

property log: `str`

Log of the filter.

Returns

`str`

Log of the filter.

class `qiuwenbot.filter.filter.TextReplaceFilter(pattern: str, repl: str)`

Bases: `Filter`

Filter to replace texts.

Parameters

pattern

[str] Pattern to replace.

repl

[str] Replacement.

Attributes

log
Log of the filter.

Methods

<i>filter</i> (text)	Filter text.
----------------------	--------------

filter(text: str) → str

Filter text.

Parameters

text
[str] Text to filter.

Returns

str
Filtered text.

property log: str

Log of the filter.

Returns

str
Log of the filter.

qiuwenbot.filter.filter.**register_filter**(cls: Filter) → Filter

Return a decorator to register filter.

The filter should not have any parameters in its constructor.

Parameters

cls
[Filter] Filter to register.

Returns

Filter
Registered filter.

Examples

```
>>> @register_filter()
... class Filter1(Filter):
...     pass
```

qiuwenbot.filter.gov module

class qiuwenbot.filter.gov.CNGovFilter

Bases: *TextReplaceFilter*

Filter to fix the Chinese government terms.

Attributes

log

Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

property `log`: `str`

Log of the filter.

Returns

`str`

Log of the filter.

qiuwenbot.filter.history module

class qiuwenbot.filter.history.FakeManchukuoFilter

Bases: *TextReplaceFilter*

Filter for Fake Manchukuo authorities.

Attributes

log

Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

property `log`: `str`

Log of the filter.

Returns

`str`

Log of the filter.

class qiuwenbot.filter.history.FakeWangFilter

Bases: *TextReplaceFilter*

Filter for Fake Wang authorities.

Attributes

log
Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

property log: `str`
Log of the filter.

Returns

`str`
Log of the filter.

qiuwenbot.filter.hk module

class `qiuwenbot.filter.hk.HKReunificationFilter`

Bases: `TextReplaceFilter`

Filter to fix the Hong Kong Reunification terms.

Attributes

log
Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

property log: `str`
Log of the filter.

Returns

`str`
Log of the filter.

qiuwenbot.filter.roc_flag module

class `qiuwenbot.filter.roc_flag.ReplaceROCyear`

Bases: `Filter`

Filter to replace ROC flag from a string.

Parameters

pattern
[str] Pattern to replace.

repl
[str] Replacement.

Attributes

log
Log of the filter.

Methods

<i>filter</i> (text)	Filter text.
----------------------	--------------

filter(text: str) → str
Filter text.

Parameters

text
[str] Text to filter.

Returns

str
Filtered text.

property log: str
Log of the filter.

Returns

str
Log of the filter.

qiuwenbot.filter.roc_year module

class qiuwenbot.filter.roc_year.ReplaceROCyear

Bases: *Filter*

Filter to replace ROC year from a string.

Parameters

pattern
[str] Pattern to replace.

repl
[str] Replacement.

Attributes

log
Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

filter(*text*: *str*) → *str*

Filter text.

Parameters

text
[*str*] Text to filter.

Returns

str
Filtered text.

property log: *str*

Log of the filter.

Returns

str
Log of the filter.

qiuwenbot.filter.tw module

class qiuwenbot.filter.tw.TWJPFilter

Bases: *TextReplaceFilter*

Filter to fix the Japanese authorities.

Attributes

log
Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

property log: *str*

Log of the filter.

Returns

str
Log of the filter.

class qiuwenbot.filter.tw.TWLeaderFilter

Bases: *TextReplaceFilter*

Filter to fix the leader name in the Taiwan area.

Attributes

log
Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

property log: `str`
Log of the filter.

Returns

`str`
Log of the filter.

class `qiuwenbot.filter.tw.TWNameFilter1`

Bases: *TextReplaceFilter*

Filter to fix the name of the Taiwan area.

Attributes

log
Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

property log: `str`
Log of the filter.

Returns

`str`
Log of the filter.

class `qiuwenbot.filter.tw.TWNameFilter2`

Bases: *TextReplaceFilter*

Filter to fix the name of the Taiwan area.

Attributes

log
Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

property log: `str`

Log of the filter.

Returns

`str`

Log of the filter.

class `qiuwenbot.filter.tw.TWQingFilter`

Bases: `TextReplaceFilter`

Filter to fix the Qing authorities.

Attributes

`log`

Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

property log: `str`

Log of the filter.

Returns

`str`

Log of the filter.

class `qiuwenbot.filter.tw.TWUnivFilter1`

Bases: `TextReplaceFilter`

Filter to fix the name of unversities in the Taiwan area.

Attributes

`log`

Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

property log: `str`

Log of the filter.

Returns

str
Log of the filter.

class qiuwenbot.filter.tw.TWUnivFilter2

Bases: *TextReplaceFilter*

Filter to fix the name of unversities in the Taiwan area.

Attributes

log
Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

property log: **str**

Log of the filter.

Returns

str
Log of the filter.

class qiuwenbot.filter.tw.TWWithOthersFilter1

Bases: *TextReplaceFilter*

Filter to fix the Taiwan name when it is with other countries.

Attributes

log
Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

property log: **str**

Log of the filter.

Returns

str
Log of the filter.

class qiuwenbot.filter.tw.TWWithOthersFilter2

Bases: *TextReplaceFilter*

Filter to fix the Taiwan name when it is with other countries.

Attributes

log
Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

property log: `str`

Log of the filter.

Returns

`str`

Log of the filter.

class `qiuwenbot.filter.tw.TWWithOthersInTitleFilter`

Bases: `Filter`

Filter to fix the Taiwan name in title when it is with other countries.

Attributes

`log`

Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

filter(*text: str*) → `str`

Filter text.

Parameters

`text`

[`str`] Text to filter.

Returns

`str`

Filtered text.

property log: `str`

Log of the filter.

Returns

`str`

Log of the filter.

qiuwenbot.filter.wg module

class qiuwenbot.filter.wg.WengeFilter

Bases: *TextReplaceFilter*

Filter to add quote to Wen Ge.

Attributes

log

Log of the filter.

Methods

<code>filter(text)</code>	Filter text.
---------------------------	--------------

property `log`: `str`

Log of the filter.

Returns

`str`

Log of the filter.

qiuwenbot.task package

Tasks.

Submodules

qiuwenbot.task.duplicate module

Check duplicated page with different variants of Chinese titles, such as zh-cn and zh-hk.

class qiuwenbot.task.duplicate.CheckDuplicatedPageTask(*user*: *str*, *password*: *str*, *pages*: *dict*)

Bases: *Task*

A task to check duplicated pages.

Parameters

user

[str] Username.

password

[str] Password.

pages

[str] Pages to operate.

Methods

<code>do(page)</code>	Do the task.
<code>logging(title)</code>	Log the removing operator.
<code>submit()</code>	Submit the task.

`do(page: Page) → bool`

Do the task.

`qiuwenbot.task.duplicate.check_page(page: Page, site: Site)`

Check if a page has duplicated variants.

qiuwenbot.task.filter module

`class qiuwenbot.task.filter.FilterTask(user: str, password: str, pages: dict)`

Bases: `Task`

A task to pass filters.

Parameters

user

[str] Username.

password

[str] Password.

pages

[str] Pages to operate.

Methods

<code>do(page)</code>	Do the task.
<code>logging(title)</code>	Log the removing operator.
<code>submit()</code>	Submit the task.

`do(page: Page) → bool`

Do the task.

qiuwenbot.task.task module

`class qiuwenbot.task.task.Task(user: str, password: str, pages: dict, logging_page: str = None, summary: str = "")`

Bases: `object`

A task to be done.

Parameters

user

[str] Username.

password

[str] Password.

pages

[dict] Pages to operate.

logging_page

[str, optional] Page to log the task, by default None

summary

[str, optional] Summary of the task, by default empty string

Methods

<code>do(page)</code>	Do the task.
<code>logging(title)</code>	Log the removing operator.
<code>submit()</code>	Submit the task.

abstract `do(page: Page) → bool`

Do the task.

logging(`title: str`) → None

Log the removing operator.

Parameters**title**

[str] title of the modified page

submit()

Submit the task.

Submodules

qiuwenbot.argparse module

`qiuwenbot.argparse.normalize(data: dict) → dict`

`qiuwenbot.argparse.page_variant() → Variant`

`qiuwenbot.argparse.submit_args() → List[Argument]`

qiuwenbot.bot module

`qiuwenbot.bot.get_page(title: str, site: Site)`

Get the page with the specific title.

Parameters**title**

[str] title of the page

site

[pywikibot.Site] qiuwen site

`qiuwenbot.bot.login(user: str, password: str) → Site`

Login to qiuwen.

Parameters

user
username of the bot

password
password of the bot

Returns

pywikibot.Site
qiuwen site

qiuwenbot.qwfamily module

`class qiuwenbot.qwfamily.QiuwenFamily`

Bases: Family

Qiuwen family.

Attributes

interwiki_forward
obsolete
Old codes that are not part of the family.

shared_urlshortner_wiki

Methods

<code>apipath(code)</code>	Return path to <code>api.php</code> .
<code>base_url(code, uri[, protocol])</code>	Prefix <code>uri</code> with port and hostname.
<code>category_redirects(code[, fallback])</code>	Return list of category redirect templates.
<code>dbName(code)</code>	Return the name of the MySQL database.
<code>disambig(code[, fallback])</code>	Return list of disambiguation templates.
<code>encoding(code)</code>	Return the encoding for a specific language wiki.
<code>encodings(code)</code>	Return list of historical encodings for a specific language wiki.
<code>eventstreams_host(code)</code>	Hostname for EventStreams.
<code>eventstreams_path(code)</code>	Return path for EventStreams.
<code>from_url(url)</code>	Return whether this family matches the given url.
<code>get_address(code, title)</code>	Return the path to <code>title</code> using <code>index.php</code> with redirects disabled.
<code>get_archived_page_templates(code)</code>	Return tuple of archived page templates.
<code>get_edit_restricted_templates(code)</code>	Return tuple of edit restricted templates.
<code>hostname(code)</code>	The hostname to use for standard http connections.
<code>interface(code)</code>	Return interface to use for <code>code</code> .
<code>isPublic()</code>	Check the wiki require logging in before viewing it.
<code>linktrail(code)</code>	Return regex for trailing chars displayed as part of a link.
<code>load([fam])</code>	Import the named family.
<code>maximum_GET_length(code)</code>	Return the maximum URL length for GET instead of POST.
<code>path(code)</code>	Return path to <code>index.php</code> .
<code>post_get_convert(site, getText)</code>	Do a conversion on the retrieved text from the Wiki.
<code>pre_put_convert(site, putText)</code>	Do a conversion on the text to insert on the Wiki.
<code>protocol(code)</code>	The protocol to use to connect to the site.
<code>querypath(code)</code>	Return path to <code>query.php</code> .
<code>scriptpath(code)</code>	The prefix used to locate scripts on this wiki.
<code>shared_image_repository(code)</code>	Return the shared image repository, if any.
<code>ssl_hostname(code)</code>	The hostname to use for SSL connections.
<code>ssl_pathprefix(code)</code>	The path prefix for secure HTTP access.
<code>verify_SSL_certificate(code)</code>	Return whether a HTTPS certificate should be verified.

instance = Family("qiuwen")

isPublic()

Check the wiki require logging in before viewing it.

langs: dict[str, str] = {'zh': 'www.qiuwenbaike.cn'}

name: str | None = 'qiuwen'

The family name

protocol(*code*)

The protocol to use to connect to the site.

May be overridden to return 'http'. Other protocols are not supported.

Changed in version 8.2: https is returned instead of http.

Parameters

code – language code

Returns

protocol that this family uses

scriptpath(*code*)

The prefix used to locate scripts on this wiki.

This is the value displayed when you enter `{{SCRIPTPATH}}` on a wiki page (often displayed at `[[Help:Variables]]` if the wiki has copied the master help page correctly).

The default value is the one used on Wikimedia Foundation wikis, but needs to be overridden in the family file for any wiki that uses a different value.

Parameters

code – Site code

Raises

KeyError – code is not recognised

Returns

URL path without ending `'/'`

qiuwenbot.qwlogger module

qiuwenbot.user-config module

qiuwenbot.utils module

`qiuwenbot.utils.archive_page`(*page: Page, site: Site*) → *Page*

Archive a page.

Parameters

page: `pywikibot.Page`

page to archive

site: `pywikibot.Site`

qiuwen site

Returns

`pywikibot.Page`

page with old title

`qiuwenbot.utils.devide_parameters`(*params: str*) → `Dict[str, str]`

Devide parameters and remove subtemplate in it.

Parameters

params

[str] parameter string

Returns

`Dict[str, str]`

dict of params

`qiuwenbot.utils.get_cat_regex(name: str = '[^\n\]]+') → Pattern`

Get categories regex.

Parameters

name

[str, optional] Name or regex of the category, by default all categories.

Returns

List[str]

Categories.

`qiuwenbot.utils.get_template_regex(name: str = '[^\n\|#0-9][^\n\|#]*?', end: str = '') → Pattern`

Get templates regex.

Parameters

name

[str, optional] Name or regex of the template, by default all templates.

end

[str, optional] End of the template, by default "".

Returns

List[str]

Templates.

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

q

- qiuwenbot, 10
- qiuwenbot.argparse, 28
- qiuwenbot.bot, 28
- qiuwenbot.entrypoints, 10
- qiuwenbot.entrypoints.gui, 10
- qiuwenbot.entrypoints.submit, 11
- qiuwenbot.filter, 11
 - qiuwenbot.filter.clean_refs, 11
 - qiuwenbot.filter.common, 12
 - qiuwenbot.filter.deprecated_tags, 12
 - qiuwenbot.filter.expired_templates, 13
 - qiuwenbot.filter.filter, 15
 - qiuwenbot.filter.gov, 18
 - qiuwenbot.filter.history, 18
 - qiuwenbot.filter.hk, 19
 - qiuwenbot.filter.roc_flag, 19
 - qiuwenbot.filter.roc_year, 20
 - qiuwenbot.filter.tw, 21
 - qiuwenbot.filter.wg, 26
- qiuwenbot.qwfamily, 29
- qiuwenbot.qwlogger, 31
- qiuwenbot.task, 26
 - qiuwenbot.task duplicate, 26
 - qiuwenbot.task.filter, 27
 - qiuwenbot.task.task, 27
- qiuwenbot.utils, 31

A

achieve_page() (in module qiuwenbot.utils), 31

C

check_page() (in module qiuwenbot.task.duplicate), 27

CheckDuplicatedPageTask (class in qiuwenbot.task.duplicate), 26

CleanRefsFilter (class in qiuwenbot.filter.clean_refs), 11

CNGovFilter (class in qiuwenbot.filter.gov), 18

D

devide_parameters() (in module qiuwenbot.utils), 31

do() (qiuwenbot.task.duplicate.CheckDuplicatedPageTask method), 27

do() (qiuwenbot.task.filter.FilterTask method), 27

do() (qiuwenbot.task.task.Task method), 28

E

end:

pages[new]/end (Argument), 9

environment variable

QIUWENBOT_PASSWORD, 7

F

FakeManchukuoFilter (class in qiuwenbot.filter.history), 18

FakeWangFilter (class in qiuwenbot.filter.history), 18

Filter (class in qiuwenbot.filter.filter), 15

filter() (qiuwenbot.filter.clean_refs.CleanRefsFilter method), 11

filter() (qiuwenbot.filter.expired_templates.RemoveExpiredTemplateFilter method), 15

filter() (qiuwenbot.filter.filter.Filter method), 15

filter() (qiuwenbot.filter.filter.FilterChain method), 16

filter() (qiuwenbot.filter.filter.TextReplaceFilter method), 17

filter() (qiuwenbot.filter.roc_flag.ReplaceROCyear method), 20

filter() (qiuwenbot.filter.roc_year.ReplaceROCyear method), 21

filter() (qiuwenbot.filter.tw.TWWithOthersInTitleFilter method), 25

FilterChain (class in qiuwenbot.filter.filter), 16

FilterTask (class in qiuwenbot.task.filter), 27

G

get_cat_regex() (in module qiuwenbot.utils), 31

get_comment() (in module qiuwenbot.filter.common), 12

get_page() (in module qiuwenbot.bot), 28

get_template_regex() (in module qiuwenbot.utils), 32

H

HKReunificationFilter (class in qiuwenbot.filter.hk), 19

I

instance (qiuwenbot.qwfamily.QiuwenFamily attribute), 30

isPublic() (qiuwenbot.qwfamily.QiuwenFamily method), 30

L

langs (qiuwenbot.qwfamily.QiuwenFamily attribute), 30

log (qiuwenbot.filter.clean_refs.CleanRefsFilter property), 11

log (qiuwenbot.filter.deprecated_tags.RemoveTagFilter property), 13

log (qiuwenbot.filter.expired_templates.RemoveExpiredTemplateFilter property), 15

log (qiuwenbot.filter.filter.Filter property), 15

log (qiuwenbot.filter.filter.FilterChain property), 16

log (qiuwenbot.filter.filter.TextReplaceFilter property), 17

log (qiuwenbot.filter.gov.CNGovFilter property), 18

log (qiuwenbot.filter.history.FakeManchukuoFilter property), 18

log (qiuwenbot.filter.history.FakeWangFilter property), 19

log (qiuwenbot.filter.hk.HKReunificationFilter property), 19

log (qiuwenbot.filter.roc_flag.ReplaceROCyear property), 20

log (*qiuwenbot.filter.roc_year.ReplaceROCYear* property), 21
 log (*qiuwenbot.filter.tw.TWJPFilter* property), 21
 log (*qiuwenbot.filter.tw.TWLeaderFilter* property), 22
 log (*qiuwenbot.filter.tw.TWNameFilter1* property), 22
 log (*qiuwenbot.filter.tw.TWNameFilter2* property), 23
 log (*qiuwenbot.filter.tw.TWQingFilter* property), 23
 log (*qiuwenbot.filter.tw.TWUnivFilter1* property), 23
 log (*qiuwenbot.filter.tw.TWUnivFilter2* property), 24
 log (*qiuwenbot.filter.tw.TWWithOthersFilter1* property), 24
 log (*qiuwenbot.filter.tw.TWWithOthersFilter2* property), 25
 log (*qiuwenbot.filter.tw.TWWithOthersInTitleFilter* property), 25
 log (*qiuwenbot.filter.wg.WengeFilter* property), 26
 logging() (*qiuwenbot.task.task.Task* method), 28
 login() (*in module qiuwenbot.bot*), 28

M

module

- qiuwenbot, 10
- qiuwenbot. argparse, 28
- qiuwenbot. bot, 28
- qiuwenbot. entrypoints, 10
- qiuwenbot. entrypoints. gui, 10
- qiuwenbot. entrypoints. submit, 11
- qiuwenbot. filter, 11
- qiuwenbot. filter. clean_refs, 11
- qiuwenbot. filter. common, 12
- qiuwenbot. filter. deprecated_tags, 12
- qiuwenbot. filter. expired_templates, 13
- qiuwenbot. filter. filter, 15
- qiuwenbot. filter. gov, 18
- qiuwenbot. filter. history, 18
- qiuwenbot. filter. hk, 19
- qiuwenbot. filter. roc_flag, 19
- qiuwenbot. filter. roc_year, 20
- qiuwenbot. filter. tw, 21
- qiuwenbot. filter. wg, 26
- qiuwenbot. qwfamily, 29
- qiuwenbot. qwlogger, 31
- qiuwenbot. task, 26
- qiuwenbot. task. duplicate, 26
- qiuwenbot. task. filter, 27
- qiuwenbot. task. task, 27
- qiuwenbot. utils, 31

N

name (*qiuwenbot.qwfamily.QiuwenFamily* attribute), 30

name:

- pages[link]/name (*Argument*), 9
- pages[page]/name (*Argument*), 10

namespace:

- pages[all]/namespace (*Argument*), 9
- pages[link]/namespace (*Argument*), 9
- pages[new]/namespace (*Argument*), 9

normalize() (*in module qiuwenbot. argparse*), 28

P

page_variant() (*in module qiuwenbot. argparse*), 28

pages (*Argument*)

- pages:, 8

pages/type (*Argument*)

- type:, 8

pages:

- pages (*Argument*), 8

pages[all]/namespace (*Argument*)

- namespace:, 9

pages[all]/restart (*Argument*)

- restart:, 9

pages[link]/name (*Argument*)

- name:, 9

pages[link]/namespace (*Argument*)

- namespace:, 9

pages[new]/end (*Argument*)

- end:, 9

pages[new]/namespace (*Argument*)

- namespace:, 9

pages[new]/start (*Argument*)

- start:, 9

pages[page]/name (*Argument*)

- name:, 10

protocol() (*qiuwenbot.qwfamily.QiuwenFamily* method), 30

Q

qiuwenbot

- module, 10

qiuwenbot. argparse

- module, 28

qiuwenbot. bot

- module, 28

qiuwenbot. entrypoints

- module, 10

qiuwenbot. entrypoints. gui

- module, 10

qiuwenbot. entrypoints. submit

- module, 11

qiuwenbot. filter

- module, 11

qiuwenbot. filter. clean_refs

- module, 11

qiuwenbot. filter. common

- module, 12

qiuwenbot. filter. deprecated_tags

- module, 12

qiuwenbot. filter. expired_templates

module, 13
 qiuwenbot.filter.filter
 module, 15
 qiuwenbot.filter.gov
 module, 18
 qiuwenbot.filter.history
 module, 18
 qiuwenbot.filter.hk
 module, 19
 qiuwenbot.filter.roc_flag
 module, 19
 qiuwenbot.filter.roc_year
 module, 20
 qiuwenbot.filter.tw
 module, 21
 qiuwenbot.filter.wg
 module, 26
 qiuwenbot.qwfamily
 module, 29
 qiuwenbot.qwlogger
 module, 31
 qiuwenbot.task
 module, 26
 qiuwenbot.task.duplicate
 module, 26
 qiuwenbot.task.filter
 module, 27
 qiuwenbot.task.task
 module, 27
 qiuwenbot.utils
 module, 31
 QIUWENBOT_PASSWORD, 7
 QiuwenFamily (class in qiuwenbot.qwfamily), 29

R

register_filter() (in module qiuwenbot.filter.filter),
 17
 RemoveExpiredCurrentFilter (class in qiuwen-
 bot.filter.expired_templates), 13
 RemoveExpiredDeadFilter (class in qiuwen-
 bot.filter.expired_templates), 13
 RemoveExpiredDeadFilter2 (class in qiuwen-
 bot.filter.expired_templates), 14
 RemoveExpiredDeadFilter3 (class in qiuwen-
 bot.filter.expired_templates), 14
 RemoveExpiredTemplateFilter (class in qiuwen-
 bot.filter.expired_templates), 14
 RemoveMapframeFilter (class in qiuwen-
 bot.filter.deprecated_tags), 12
 RemoveScoreFilter (class in qiuwen-
 bot.filter.deprecated_tags), 12
 RemoveTagFilter (class in qiuwen-
 bot.filter.deprecated_tags), 12

RemoveTimelineFilter (class in qiuwen-
 bot.filter.deprecated_tags), 13
 ReplaceROCYear (class in qiuwenbot.filter.roc_flag), 19
 ReplaceROCYear (class in qiuwenbot.filter.roc_year), 20
 restart:
 pages[all]/restart (Argument), 9

S

scriptpath() (qiuwenbot.qwfamily.QiuwenFamily
 method), 31
 start:
 pages[new]/start (Argument), 9
 start_dpgui() (in module qiuwenbot.entrypoints.gui),
 10
 submit() (in module qiuwenbot.entrypoints.submit), 11
 submit() (qiuwenbot.task.task.Task method), 28
 submit_args() (in module qiuwenbot.argparse), 28

T

task (Argument)
 task:, 10
 Task (class in qiuwenbot.task.task), 27
 task:
 task (Argument), 10
 TextReplaceFilter (class in qiuwenbot.filter.filter), 16
 TWJPFfilter (class in qiuwenbot.filter.tw), 21
 TWLeaderFilter (class in qiuwenbot.filter.tw), 21
 TWNameFilter1 (class in qiuwenbot.filter.tw), 22
 TWNameFilter2 (class in qiuwenbot.filter.tw), 22
 TWQingFilter (class in qiuwenbot.filter.tw), 23
 TWUnivFilter1 (class in qiuwenbot.filter.tw), 23
 TWUnivFilter2 (class in qiuwenbot.filter.tw), 24
 TWWithOthersFilter1 (class in qiuwenbot.filter.tw), 24
 TWWithOthersFilter2 (class in qiuwenbot.filter.tw), 24
 TWWithOthersInTitleFilter (class in qiuwen-
 bot.filter.tw), 25
 type:
 pages/type (Argument), 8

U

user (Argument)
 user:, 8
 user:
 user (Argument), 8

W

WengeFilter (class in qiuwenbot.filter.wg), 26