CampBot Documentation

Release latest

Mar 07, 2022

Contents

1 Installation 3

2 Command line interface 5

3 Python recipes 9

4 API reference 15

CampBot is a Python framework for build upon camptocamp.org API.

CHAPTER 1

Installation

1.1 Requirements

- Working Python 3.6 or higher

and, optionnaly if you need to modify documents :

- a camptocamp.org account for editing documents ;
- a demov6camptocamp.org account for testing editions.

1.2 Installation

```
pip install campbot
```

1.3 Test installation

In command line interface, type :

```
campbot
```

This command will display help for command line API.

In python, simply import CampBot class :

```
from campbot import CampBot
```

CHAPTER 2

Command line interface

Here isthe help you get by typing `campbot` in you command line tool :

```
CampBot, Python bot framework for camptocamp.org

Usage:
  campbot clean_rc <days> <lang> [--login=<login>] [--password=<password>] [--delay=
↪<seconds>] [--batch]
  campbot report_rc <days> <lang> [--login=<login>] [--password=<password>] [--delay=
↪<seconds>] [--batch]
  campbot clean <url_or_file> <lang> [--login=<login>] [--password=<password>] [--
↪delay=<seconds>] [--batch] [--bbcode]
  campbot contribs [--out=<filename>] [--starts=<start_date>] [--ends=<end_date>] [--
↪delay=<seconds>]
  campbot export <url> [--out=<filename>] [--delay=<seconds>]


Options:
  --login=<login>           Bot login
  --password=<password>     Bot password
  --batch                   Batch mode, means that no confirmation is required before␣
↪saving
                            Use very carefully!
  --delay=<seconds>         Minimum delay between each request. Default : 3 seconds
  --bbcode                  Clean old BBCode in markdown
  --out=<filename>          Output file name. Default value will depend on process


Commands:
  report_rc     Make quality report on recent changes.
  clean_rc      Clean recent changes.
  clean         Clean documents.
                <url_or_file> is like https://www.camptocamp.org/routes#a=523281, or,␣
↪simplier, routes#a=523281.
                filename is also accepted, and must be like :
```

(continues on next page)

(continued from previous page)

```
                123 | r
                456 | w
                <lang> is a lang identifiers, like fr for french.
  contribs      Export all contribution in a CSV file. <start_date> and <end_date>␣
↪are like 2018-05-12
  export        Export all documents in a CSV file.
                <url> is like https://www.camptocamp.org/outings#u=2, or, simplier,␣
↪outings#u=2
```

2.1 Export a list of document

Export to a csv file all documents given by camptocamp URL

2.1.1 Command line

```
campbot export <url> [--out=<filename>] [--delay=<seconds>]
```

- `<url>` is a camptocamp url, like https://www.camptocamp.org/routes#a=523281
- `<filename>` is the output file. By default, it will be `outings.csv` for outings, `routes.csv` for routes...
- `<seconds>`, numerical, is the delay between each request. 3s by default.

2.1.2 Output

here is a sample of output:

```
date_start;date_end;title;equipement_rating;global_rating;height_diff_up;rock_free_␣
↪rating;condition_rating;elevation_max;img_count;quality;activities
2017-11-20;2017-11-20;Calanque de Morgiou - Le Cancéou : Aven du Cancéou - Prends moi␣
↪sec au-dessus du lagon bleu;;D+;100;5c;excellent;None;0;draft;rock_climbing
2017-11-19;2017-11-19;Calanque de Sormiou - Dièdre Guem : Voie NTD;;TD+;140;6c+;
↪excellent;N
```

2.2 Export whatsnew page

This command will load all contributions made in the last 24 hours

2.2.1 Command line

```
campbot contribs
```

2.2.2 Optional arguments

- `--starts=2017-12-07` : will export all contributions after this date (included)
- `--ends=2017-12-07` : will export all contributions before this date (excluded)

- `--out=data.csv` : out file name, default value is contributions.csv

2.2.3 Output

here is a sample of output:

```
timestamp;type;document_id;version_id;document_version;title;quality;username;lang
2017-12-21T21:49:41.363647+00:00;routes;293549;1738922;4;Escalades à Presles;medium;
↪charles b;fr
2017-12-20T21:49:41.363647+00:00;routes;123;123;4;Escalades à Presles;medium;munch;fr
```

2.3 Clean a set of documents

This command line will process all clean processors to objects given by a camptocamp url.

2.3.1 Command line

```
campbot clean <url> <lang> --login=<login> --password=<password> [--delay=<seconds>]␣
↪[--bbcode]
```

2.3.2 Options and arguments

- `<url>` is like https://www.camptocamp.org/routes#w=940468 : all routes associated to waypoint 940468 will be cleaned. Shorthand `routes#w=940468` is accepted.
- `<lang>` is a lang identifier, like fr or de. Clean procedure will impacts only this lang.

2.3.3 Clean processors

- https://www.camptocamp.org/articles/996571
- Some letter capitalization
- Spaces between numbers and units

2.3.4 Good advice

`--delay` is the time between each API request. By default, it's 3s, which is very low. As it's time consuming, it's preferable to set `--delay` to a low value. As you will have to validate each modification, API w'ont be overloaded.

2.3.5 Sample

```
campbot clean https://www.camptocamp.org/routes#w=940468 fr --login=rabot --
↪password=fake_pwd --delay=0.1
```

2.3.6 bbcode

This option clean old good BBCode tags with their markdown equivalents. No more usefull, except for your outings :

```
campbot clean outings#u=123 fr --login=your_login --password=fake_pwd --bbcode
```

Warning: `123` must be your user numerical id, you can find it in your home page's URL.

2.4 Quality report on recent changes

This procedure will check that last day modifications pass these tests

- History is filled
- No big deletions
- Pitch type is set
- Warning for new users
- all patterns present in the first message of topic where report should be posted

Report will be posted on https://forum.camptocamp.org/t/topoguide-verifications-automatiques/201480

2.4.1 Command line

```
campbot report_rc <days> <lang> <thread_url> [--login=<login>] [--password=<password>
↪] [--delay=<seconds>]
```

2.4.2 Arguments and options

- `<days>` : Number of day to check Let says that we are on 17 june 2018, 12h. if `<days>` is `2`, the process will run on all contributions made from `2018-06-15 00:00:00` to `2018-06-16 23:59:59`
- `<lang>` : Only contributions made on this lang will be checked
- `<thread_url>` : Thread's URL where the report will be published
- `<login>` : bot's login
- `<password>` : bot's password
- `<delay>` : delay, in seconds between each request. By defaut, 3 seconds

Warning: This process is quite long. So, please consider this two points :

- Execute it when camptocamp is not overloaded
- and use a delay of 0.1 seconds.

CHAPTER 3

Python recipes

How to.... ?

As a good short example will always be better than a long explanation, here is some code sample. Inspire yourself with them.

3.1 Get documents

Here is a code that load every single documents based on a filter. Corresponding Camptocamp URL is https://www.camptocamp.org/routes#a=14405&act=mountain_climbing%252Crock_climbing

```
from campbot import CampBot

bot = CampBot(use_demo=True)

# Let get all routes inside area 14405 (French Jura),
# AND where activity is rock_climbing OR mountain_climbing

filters = {
    "a": 14405,
    "act":["rock_climbing", "mountain_climbing"]
}

for route in bot.wiki.get_routes(filters):
    print(route)
```

`bot.wiki` object has also theese functions :

- `bot.wiki.get_areas(filters)`
- `bot.wiki.get_outings(filters)`
- `bot.wiki.get_waypoints(filters)`
- `bot.wiki.get_xreports(filters)`

Warning: Due to Camptcamp database limitations, you won't be able to load more than 10,000 documents with this method. Please use a more precise filter if you need to crawl a big amount of data. On the other hand, it should be more efficient to ask to Camptocamp association a static dump if you need to download a consequent part of C2C data.

3.2 Search documents that contains...

Based on *get document sample*, here is the code to search for text. This code is very slow, but may do the job for a small set of documents

```
from campbot import CampBot

bot = CampBot(use_demo=True)

# Here is the text to search
needle_text = "aiguille dans une botte de foin"

# Use a pre-filter, otherwise, it will never end
filters = {
    "a": 14405,
    "act":["rock_climbing", "mountain_climbing"]
}

for route in bot.wiki.get_routes(filters):
    locale = route.get_locale("fr") # we will search only french text

    if locale: # document may not contain french version
        for field in locale:
            if locale[field] is not None: # field may be set to None
                if needle_text in locale[field]:
                    print(route)
```

3.3 Add a rock type to a list of document

This script will add limestone rock type ("calcaire in french") on all routes associated to waypoint `107702` (Verdon in France, which is actually a limestone area) :

```
from campbot import CampBot

bot = CampBot(use_demo=True)
bot.login("bot_login", "bot_password")

filters = {"w": 107702}

for route in bot.wiki.get_routes(filters):
    route["rock_types"] = route.get("rock_types", []) or []
    if "calcaire" not in route["rock_types"]:
        route["rock_types"].append("calcaire")
        route.save("Only limestone in Verdon")
```

3.4 Get user from route

It can be usefull, when you want to discuss about a point on a route, to mention all users that knows this route. This code will display all forum names, 20 per rows (Discourse limit), with a preceding `@`.

```
from campbot import CampBot

bot = CampBot(use_demo=True)

users = bot.get_users_from_route(56048)

for sub_users in [users[i:i + 20] for i in range(0, len(users), 20)]:
    print(", ".join(["@" + user["forum_username"] for user in sub_users]))
```

3.5 Update a locale field

This script will update remarks field.

```
import sys
from campbot import CampBot


bot = CampBot()
bot.login("botname", "botpass")

remarks="!! VTNO rocks!"
update_comment = "Remarque gestion par VTNO"

filters = {
    "w": 107049,
    "act":["rock_climbing"]
}

for route in bot.wiki.get_routes(filters):
    locale = route.get_locale("fr")

    if locale: # document may have not fr locale : skip it.
        if not locale.remarks: # remarks is none, or empty string
            locale.remarks = remarks
            route.save(update_comment)

        elif "VTNO" not in locale.remarks:
            # We consider that if remarks field still containes "VTNO",
            # then locale is still processed
            locale.remarks = locale.remarks + "\n\n" + remarks
            route.save(update_comment)
```

3.6 Export data as CSV

CSV (comma separated values) is a convenient way to export data for later analysis on spreadsheet softwares (Excel...). Here is a code that exports some informations from images associated to an article

```
from campbot import CampBot

bot = CampBot(min_delay=0.01)

article = bot.wiki.get_article(1058594)  # Concours Photo Sophie 2018


def output(*args):
    print(";".join(map(str, args)))


output("image", "licence", "creator", "creator_id", "Associated route count",
    "Associated waypoint count")

for image in article.associations.images:
    image = bot.wiki.get_image(image.document_id)
    output(image.get_url(), image.image_type, image.creator["name"],
        image.creator["user_id"], len(image.associations.routes),
        len(image.associations.waypoints))
```

3.7 Create an area

There is no UI for adding new areas. But API entry point is up and running!

```
import json
from campbot import CampBot


def create_area(bot, title, geom_detail):
    result = bot.wiki.post(
        "/areas",
        {
            "geometry": {"geom": None, "geom_detail": json.dumps(geom_detail),},
            "type": "a",
            "quality": "medium",
            "area_type": "admin_limits",
            "available_langs": ["fr"],
            "locales": [
                {"lang": "fr", "summary": None, "title": title, "description": None,}
            ],
        },
    )

    print(result)

bot = CampBot()
bot.login("moderator_login", "password")

geom_detail = {
    "coordinates": [
        [
            [5012713.121726842, -1417626.9752307558],
            [5014720.969939139, -1419322.4914989625],
            [5016862.674699049, -1421910.3847504659],
            [5012713.121726842, -1417626.9752307558],  # this point must be the same␣
↪as the first one
```

(continues on next page)

(continued from previous page)

```
        ]
    ],
    "type": "Polygon",
}

create_area(bot, "New area", geom_detail)
```

3.8 Check that Sophie picture contest criteria has been respected

The rules are :

- a picture can't be resubmitted
- a participant cannot submit more than 3 pictures

```
from collections import defaultdict
from campbot import CampBot

bot = CampBot(min_delay=0.01)

article_ids = [
    187913,
    237549,
    300413,
    374949,
    465897,
    555996,
    673796,
    809627,
    937458,
    1058594,
    1058594,
]

print("Load images of previous challenges", end ="")
images_id = {}
for article_id in article_ids:
    article = bot.wiki.get_article(article_id)
    for image in article.associations.images:
        images_id[image.document_id] = article_id
print(" - Ok")

print("Check that no image has been submitted in a previous challenges", end="")
article = bot.wiki.get_article(1251594)
for image in article.associations.images:
    if image.document_id in images_id:
        raise Exception(f"Image {image.document_id} has been submitted on article
↪{images_id[image.document_id]}")
print(" - Ok")

print("Check that no user has submitted more than 3 images:")
users = defaultdict(list)
for image in article.associations.images:
    full_image = bot.wiki.get_image(image.document_id)
    users[full_image.creator['user_id']].append(image.document_id)
```

(continues on next page)

(continued from previous page)

```
for user, images in users.items():
    if len(images) > 3:
        print(f"User {user} has submitted {len(images)} images: {images}")
```

3.9 Get routes from books

Get all routes associated to a set of books

```
"""
Usage:
    python get_routees.py 14637 14538 179160 14652 > results.txt
"""

import campbot
import sys

bot = campbot.CampBot()

books_ids = sys.argv[1:]

routes = []
for books_id in books_ids:
    book = bot.wiki.get_book(books_id)
    routes += book["associations"]["routes"]

def get_title(r):
    return r.get_title("fr") or r.get_title("it") or r.get_title("de") or r.get_title(
↪"es")

routes = [campbot.objects.Route(bot, r) for r in {r["document_id"]: r for r in routes}
↪.values()]
routes = sorted(routes, key=get_title)

for r in routes:
    print(f'1. {r.get_url()} : {get_title(r)}')
```

CHAPTER 4

API reference

Warning: Normally, all public methods in Python sould be named normally, and all private methods should be preceded by an underscore. As `campbot` is quite new, and very moving, please consider that if a object/method is NOT documented in this section, then you must not use it. Or at your own risk.

4.1 CampBot class

4.1.1 Properties

- `bot.wiki` : a *WikiBot instance*
- `bot.forum` : a *ForumBot instance*
- `bot.moderator` : Boolean, True if logged with a moderator account

4.2 WikiBot class

4.2.1 About document_type/constructor parameters

Some function of WikiBot will ask you to specify which kind of object you want to request. There is two way to specify it :

1. Set document_type argument with a letter (see bellow the mapping)
2. Set constructor argument, see campbot.objects module

Mapping :

- "u": WikiUser
- "a": Area

- “w”: Waypoint
- “o”: Outing
- “i”: Image
- “m”: Map
- “x”: Xreport
- “c”: Article
- “b”: Book
- “r”: Route

Example : this two line are equivalents

```
from campbot import Campbot, objects

bot = Campbot(user_demo=True)

bot.wiki.get_documents(filters, document_type='r')
bot.wiki.get_documents(filters, constructor=objects.Route)
```

4.2.2 About filters

Some function that return list of objects have a `filters` argument. It’s a key value dictionary. Simply see camptocamp URL to understand how to fill it.

4.2.3 API

4.3 ForumBot class

4.4 Data objects

4.5 Document processors