easyadwords Documentation

Release 0.1.3

Daniel Poon

September 08, 2016

Contents

- genindex
- modindex
- search

Contents:

CHAPTER 1

easyadwords

User friendly wrapper for Google AdWords

- Free software: Apache Software License 2.0
- Documentation: https://easyadwords.readthedocs.io.

1.1 Features

- Access most reporting functions of the powerful AdWords API (relatively) painlessly

1.2 Installation

```
$ pip install easyadwords
```

1.3 Authors

1.3.1 Credits

Development Lead

- Daniel Poon <daniel.poon.wenjie@gmail.com>

Contributors

None yet. Why not be the first?

1.4 Contributing

1.4.1 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

Types of Contributions

Report Bugs

Report bugs at https://github.com/danielpoonwj/easyadwords/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

Write Documentation

easyadwords could always use more documentation, whether as part of the official easyadwords docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/danielpoonwj/easyadwords/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

Get Started!

Ready to contribute? Here's how to set up *easyadwords* for local development.

1. Fork the *easyadwords* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/easyadwords.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv easyadwords
$ cd easyadwords/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

6. Submit a pull request through the GitHub website.

Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.

1.5 History

1.5.1 History

0.1.3 (2016-09-08)

- Retry with incremental backoff logic applied to most functions

0.1.2 (2016-08-29)

- Added get_all_account_info to AdwordsUtility
- Added date_range to utils

0.1.1 (2016-08-29)

- Download report as GZIPPED_CSV, then unzip and load locally.

0.1.0 (2016-08-26)

- First release on PyPI.

1.6 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

CHAPTER 2

Installation

2.1 Stable release

To install easyadwords, run this command in your terminal:

```
$ pip install easyadwords
```

This is the preferred method to install easyadwords, as it will always install the most recent stable release.

If you don’t have pip installed, this Python installation guide can guide you through the process.

2.2 From sources

The sources for easyadwords can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/danielpoonwj/easyadwords
```

Or download the tarball:

```
$ curl  -OL https://github.com/danielpoonwj/easyadwords/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER 3

Usage

This package wraps the official googleads package to focus on simplifying the retrieval of Reports and Services through the API.

3.1 Authentication

To be granted access to AdWords' API, follow the steps listed here: https://developers.google.com/adwords/api/docs/guides/signup

Once you have received your developer token, create **googleads.yaml** in a secure location and fill in the following fields.

```
adwords:
  developer_token:
  user_agent:
  client_id:
  client_secret:
  refresh_token:
# client_customer_id:
```

This credentials file would then be used to authenticate the AdwordsUtility. Specifying *client_customer_id* in the file is optional, but it has to be either written explicitly in the file or input when creating the AdwordsUtility object.

```
from easyadwords import AdwordsUtility
adwords_obj = AdwordsUtility(credential_path='path/to/credentials/googleads.yaml')
```

3.2 Caveats

If you're using other packages that rely on oauth2client, googleads dependency specifies **oauth2client<2.0.0,>=1.5.2**. You may have to update oauth2client separately if you're dependant on a more recent version.

Documentation

4.1 Adwords

4.1.1 AdwordsUtility

class `easyadwords.`**`AdwordsUtility`**(*credential_path*, *client_customer_id=None*, *service_version=None*, *max_retries=3*)

Initialize new utility object for interacting with Adwords.

Configuration/authorization is determined from googleads.yaml (credential_path).

Parameters

- **`credential_path`** – Path to googleads.yaml
- **`client_customer_id`** – Default customer_id, would override that stated in credential_path.
- **`service_version`** – If set, get specific version. Else, get the latest available version. **NOTE** Check change logs for APIs and googleads client before upgrading or switching report versions.

`change_client_customer_id`(**args*, ***kwargs*)

Set new client_customer_id.

`get_report_fields`(**args*, ***kwargs*)

Get details about report fields.

Parameters

- **`report_type`** – Reference: https://developers.google.com/adwords/api/docs/appendix/reports#report-types
- **`serialize`** – Convert to dictionary.

Returns list of dictionaries or SOAP responses depending on serialize option.

`get_service`(**args*, ***kwargs*)

General purpose function for getting any service listed here: https://developers.google.com/adwords/api/docs/reference/

Parameters

- **`service_name`** – Name of service
- **`selector`** –

- **iterate_pages** –
- **serialize** –

Returns

list_account_labels()

Convenience function for AccountLabelService with predefined options.

Returns list of dictionaries

list_accounts(*fields=None*, *predicates=None*, *include_hidden=False*, *include_mcc=False*, *serialize=True*)

Convenience function for ManagedCustomerService with predefined options.

Parameters

- **predicates** (*list of dictionaries representing Predicate objects*) – Predicate objects for filtering data.
- **include_hidden** – Include hidden accounts in results.
- **include_mcc** – Include MCC in results.
- **serialize** – Convert to dictionary.

Returns list of dictionaries or SOAP responses depending on serialize option.

get_report(**args*, ***kwargs*)

Downloads and cleans report.

Field Examples:

Renaming field:

{'name': 'Ctr', 'alias': 'ctr'}

Custom Cleaning:

NOTE - simplest implementation would be using a lambda function as shown below.

{'name': 'Ctr', 'alias': 'ctr', 'cleaning': lambda x: float(str(x).replace('%', '').strip())}

Additional Field Examples:

Prepending field "updated_at":

{'name': 'updated_at', 'value': datetime.now(), 'prepend'=True}

Parameters

- **start_date** (*datetime*) – Reporting start date.
- **end_date** (*datetime*) – Reporting end date.
- **report_type** – Reference: https://developers.google.com/adwords/api/docs/appendix/reports#report-types
- **fields** (*list of dictionaries*) – Fields within report.
- **additional_fields** (*list of dictionaries*) – New fields to add. **Only supports static values, not functions or references to other columns.**
- **predicates** (*list of dictionaries representing Predicate objects*) – Predicate objects for filtering data.
- **client_customer_id** – Overwrite set client_customer_id when downloading report.
- **include_zero_impressions** – **Check compatibility with report type**

Returns Generator object for cleaned report

get_all_account_info(*start_date*, *end_date*)

Convenience function wrapping ACCOUNT_PERFORMANCE_REPORT to get and parse accounts info. Can be used to subsequently filter out accounts without any activity for specific days.

Parameters

- **start_date** (*datetime object*) – Start date
- **end_date** – End date

Returns Dictionary structured by account id > date > metrics

4.1.2 Misc Classes/Functions

easyadwords.utils.**serialize_soap_resp**(*resp*)

Convert Adwords SOAP response to serializable dict

Parameters resp – SOAP response

Returns Dictionary representation of response

easyadwords.utils.**date_range**(*start*, *end*, *ascending=True*, *date_format='%Y-%m-%d'*)

Simple datetime generator for dates between start and end (inclusive).

Parameters

- **start** (*datetime object or string representation of datetime.*) – Date to start at.
- **end** (*datetime object or string representation of datetime.*) – Date to stop at.
- **ascending** (*boolean*) – Toggle sorting of output.
- **date_format** – If input is string, denotes string datetime format to convert from.

Returns generator object for naive datetime objects

Python Module Index

Index