
clan Documentation

Release 0.2.4 (beta)

Christopher Groskopf

December 17, 2014

1	About	1
1.1	clan (Command Line ANalytics)	1
2	See clan in action	3
3	Getting started	5
3.1	Installation	5
3.2	Authentication	5
4	Usage	7
4.1	Basic usage	7
4.2	Configuration	8
4.3	Common queries	11
5	Development	13
5.1	Release process	13
6	Authors	15
7	License	17
8	Changelog	19
8.1	0.2.4	19
8.2	0.2.3	19
8.3	0.2.2	19
8.4	0.2.1	19
8.5	0.2.0	19
8.6	0.1.3	20
8.7	0.1.2	20
8.8	0.1.1	20
8.9	0.1.0	20
9	Indices and tables	21

1.1 clan (Command Line ANalytics)

A command line utility for generating Google Analytics reports that are straightforward to compare across domains, projects or pages.

Important links:

- Documentation: <http://clan.rtfid.org/>
- Repository: <https://github.com/onyxfish/clan>
- Issues: <https://github.com/onyxfish/clan/issues>

See clan in action

From NPR Visuals Book Concierge 2014 project:

- Clan report
- Clan configuration .

Getting started

3.1 Installation

3.1.1 Users

If you only want to use clan, install it this way:

```
pip install clan
```

Note: clan is intended for **researchers** and **analysts**. You will need to understand the Google Analytics API in order to use it effectively. It is not intended to generate reports for your boss.

3.1.2 Developers

If you are a developer that also wants to hack on clan, install it this way:

```
git clone git://github.com/onyxfish/clan.git
cd clan
mkvirtualenv --no-site-packages clan
pip install -r requirements.txt
python setup.py develop
```

Note: If you have a recent version of pip, you may need to run pip with the additional arguments `--allow-external argparse`.

3.2 Authentication

Before you use clan, you're going to need to setup your access to the Google Analytics API. Follow the [instructions in Google's docs](#) to register an application and create the `client_secrets.json` file.

Once you've got a `client_secrets.json` file, clan will walk you through acquiring an oAuth token:

```
clan auth
```

By default this token will be named `analytics.dat`. I suggest you move this file to `~/.clan_auth.dat`. clan will always look for the auth in that location so you will only need one copy no matter what directory you are running clan from.

4.1 Basic usage

clan has three basic uses

- Writing query results to an text or HTML report suitable for reading.
- Writing query results to a JSON file suitable for further processing.
- Generating an HTML “diff”, or change report, comparing two JSON outputs.

4.1.1 Generating a text report

To configure clan, create a YAML data file describing the analytics you want to run:

```
# Global configuration, only property-id is required
title: Commencement report
property-id: "53470309"
start-date: "2014-06-01"
prefix: "/commencement/"

# Metrics to report
queries:
  - name: Totals
    description: Top-level counts
    metrics:
      - "ga:pageviews"
      - "ga:uniquePageviews"
      - "ga:users"
      - "ga:sessions"

  - name: Totals by device category
    description: Device categories are desktop, tablet and mobile
    metrics:
      - "ga:pageviews"
      - "ga:uniquePageviews"
      - "ga:users"
      - "ga:sessions"
    dimensions:
      - "ga:deviceCategory"
    sort:
      - "-ga:pageviews"
```

Assuming this configuration is named “configuration.yml”, to produce an HTML report for this configuration you would run the following command.

```
clan report configuration.yml report.html
```

For complete documentation of this configuration, see *Configuration*.

4.1.2 Generating a JSON report

Instead of HTML you can output data in a JSON microformat suitable for diffing, archiving, visualization or further processing with other tools:

```
clan report configuration.yml report.json
```

Global configuration options, such as `start-date` can also be specified as command line arguments, allowing you to reuse a YAML configuration file for several projects. When specified, command-line arguments will always take precedence over options defined in the YAML configuration.

```
clan report --start-date 2014-05-1 --prefix /tshirt/ configuration.yml report.json
```

You can also convert an HTML report from an existing JSON report:

```
clan report analytics.json report.html
```

4.1.3 Generating a text diff

If you report on multiple projects using the same analytics, you can use clan to compare their performance:

```
clan diff a.json b.json diff.html
```

The values in the diff report columns will be:

- Absolute difference
- Percent change
- Change in percentage points

4.1.4 Generating a JSON diff

As with individual reports, diffs can also be saved as JSON for further processing:

```
clan diff a.json b.json diff.json
```

4.2 Configuration

4.2.1 Configuring with YAML

clan is configured using either YAML, command-line arguments or both.

The basic structure of the YAML configuration file is:

```
# Global configuration
title: Sample configuration
property-id: "53470309"

# A list of queries to execute
queries:

  # Individual query configuration
  - name: Totals
    description: Property-wide top-level totals.
    metrics:
      - "ga:pageviews"
      - "ga:uniquePageviews"
      - "ga:users"
      - "ga:sessions"
```

4.2.2 Global configuration

The following is a list of properties that may be specified in as global configuration. Note that these may also be specified using command line arguments. Some properties can also be specified on a per-query basis. If there is a disagreement, the values will be preferred in the following order:

1. Command-line values
2. Query configuration in YAML
3. Global configuration in YAML

title

A user-friendly title for the report.

property-id

The ID of the Google Analytics property to query.

start-date

The start date of all queries, in YYYY-MM-DD format.

end-date

The end date of all queries, in YYYY-MM-DD format. Supersedes `ndays` if both are specified.

ndays

A number of days from the start date to report on. Superseded by `end-date` if both are specified.

domain

If specified, results will be limited to URLs from this domain.

prefix

If specified, results will be limited to URLs with this prefix.

4.2.3 Per-query configuration

Individual queries support the following properties.

name

A brief name for the query..

description

A longer description of the query.

metrics

A list of Google Analytics metrics to be reported.

For details about all metrics you can report on, see the [Google Analytics Core Reporting API docs](#).

dimensions

A list of Google Analytics metrics on which to segment the data. Not that these are pairwise not hierarchical. If your query configuration looked like:

```
- name: Pageviews by device and browser
  metrics:
    - "ga:pageviews"
  dimensions:
    - "ga:deviceCategory"
    - "ga:browser"
  sort:
    - "-ga:pageviews"
```

Then your resulting report would enumerate the most popular combinations of device and browser, not the most popular devices further subdivided by most popular browser.

sort

A list of Google Analytics metrics to sort by. Prefix a value with a - to sort in descending order.

filter

A Google Analytics [query filter expression](#) to apply to the data. This will be “ANDed” together with any filters automatically generated from other configuration options such as `domain` or `prefix`.

segment

A Google Analytics [segment definition](#) to use to filter the data.

4.3 Common queries

4.3.1 Total pageviews, uniques, users, etc.

```
- name: Totals
  metrics:
    - "ga:pageviews"
    - "ga:uniquePageviews"
    - "ga:users"
    - "ga:sessions"
```

4.3.2 Device share

Get totals broken down by desktop, tablet and mobile.

```
- name: Totals by device type
  metrics:
    - "ga:pageviews"
    - "ga:uniquePageviews"
    - "ga:users"
    - "ga:sessions"
  dimensions:
    - "ga:deviceCategory"
  sort:
    - "-ga:pageviews"
```

4.3.3 Browser share

```
- name: Totals by browser
  metrics:
    - "ga:pageviews"
  dimensions:
    - "ga:browser"
  sort:
    - "-ga:pageviews"
```

4.3.4 Most viewed pages

```
- name: Top pages
  metrics:
    - "ga:pageviews"
  dimensions:
    - "ga:pagePath"
  sort:
    - "-ga:pageviews"
  max-results: 20
```

4.3.5 Top sources (referrers)

```
- name: Totals by source
  metrics:
    - "ga:pageviews"
  dimensions:
    - "ga:source"
  sort:
    - "-ga:pageviews"
```

4.3.6 Top social sources

```
- name: Totals by social network
  metrics:
    - "ga:pageviews"
  dimensions:
    - "ga:socialNetwork"
  sort:
    - "-ga:pageviews"
```

4.3.7 Page load and render times

```
- name: Performance
  metrics:
    - "ga:avgPageLoadTime"
    - "ga:avgPageDownloadTime"
    - "ga:avgDomInteractiveTime"
    - "ga:avgDomContentLoadedTime"
```

4.3.8 Time on site

```
- name: Time on site
  metrics:
    - "ga:avgSessionDuration"
```

4.3.9 Custom event count

```
- name: "Event: tweet"
  metrics:
    - "ga:totalEvents"
    - "ga:uniqueEvents"
  filter: "ga:eventAction==tweet"
```

4.3.10 Custom event value

```
- name: "Event: time-on-slide"
  metrics:
    - "ga:eventValue"
    - "ga:avgEventValue"
  filter: "ga:eventAction==time-on-slide"
```


5.1 Release process

1. Verify no high priority issues are outstanding.
2. **Ensure these files all have the correct version number:**
 - CHANGELOG
 - setup.py
 - docs/conf.py
 - clan/templates/report.html (footer)
 - clan/templates/diff.html (footer)
3. Tag the release: `git tag -a x.y.z; git push --tags`
4. Roll out to PyPI: `python setup.py sdist upload`
5. Iterate the version number in all files where it is specified. (see list above)
6. Flag the new version for building on [Read the Docs](#).
7. Wait for the documentation build to finish.
8. Flag the new release as the default documentation version.
9. Announce the release on Twitter, etc.

Authors

- Christopher Groskopf
- Tyler Fisher
- Danny DeBelius

License

The MIT License

Copyright (c) 2014 Christopher Groskopf and contributors

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.

Changelog

8.1 0.2.4

- Add example output to docs.

8.2 0.2.3

- Packaging problems.

8.3 0.2.2

- Packaging problems.

8.4 0.2.1

- Packaging problems.

8.5 0.2.0

- Rev google-api-python-client to latest version.
- Documented a release process.
- Added Tyler Fisher to AUTHORS.
- Added Danny DeBelius to AUTHORS.
- Restructure code so pip works. (#26)
- Fetch human-readable field names from Google.
- Simplify command-line usage.
- Kill text support.
- Redesign HTML output.

- Added user-configurable query description property.
- Added user-configurable report title property.
- Added support for query segment'' property.

8.6 0.1.3

- Fix lots of template bugs. (#17, #18)
- Add HTML output for reports and diffs. (#9)

8.7 0.1.2

- Add *clan diff* command. (#8)

8.8 0.1.1

- Refactored to use command structure for CLI.
- `--ndays` argument. (#10)
- Document all configuration options. (#13)
- Allow global configuration on command line. (#12)
- Fixed `.yaml` extension to be `.yml`.

8.9 0.1.0

- Initial version.

Indices and tables

- *genindex*
- *modindex*
- *search*