
clan Documentation

Release 0.1.3 (alpha)

Christopher Groskopf

December 17, 2014

1	About	1
1.1	clan (Command Line ANalytics)	1
2	Getting started	3
2.1	Installation	3
2.2	Authentication	3
3	Usage	5
3.1	Basic usage	5
3.2	Configuration	8
3.3	Common queries	10
4	Authors	13
5	License	15
6	Changelog	17
6.1	0.1.3	17
6.2	0.1.2	17
6.3	0.1.1	17
6.4	0.1.0	17
7	Indices and tables	19

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:

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

Getting started

2.1 Installation

2.1.1 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.

2.1.2 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`.

2.2 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.

Usage

3.1 Basic usage

clan has three basic uses

- Writing query results to a text or HTML report suitable for reading or emailing.
- Writing query results to a JSON file suitable for further processing.
- Generating a “diff”, or change report, comparing two sets of query results, as either text or JSON.

3.1.1 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
property-id: "53470309"
start-date: "2014-06-01"
prefix: "/commencement/"

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

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

To run this report to a JSON file, run the following command. Note that by default clan will look for a YAML file called `clan.yml`. You can override this with the `-c` option. For complete documentation of this configuration, see *Configuration*.

To produce a text report, run:

```
clan report analytics.txt
```

Here is sample output for the above configuration:

Report run 2014-06-06 with:

```
property-id: 53470309
start-date: 2014-06-01
ndays: 2
prefix: /commencement/
```

Totals

(using 89.0% of data as sample)

ga:pageviews			
88,935	100.0%	total	
ga:uniquePageviews			
60,179	100.0%	total	
ga:users			
21,244	100.0%	total	
ga:sessions			
26,817	100.0%	total	

Totals by device category

(using 89.0% of data as sample)

ga:pageviews			
64,542	72.6%	desktop	
15,403	17.3%	mobile	
8,991	10.1%	tablet	
88,936	100.0%	total	
ga:uniquePageviews			
40,966	68.1%	desktop	
12,277	20.4%	mobile	
6,936	11.5%	tablet	
60,179	100.0%	total	
ga:users			
12,838	60.4%	desktop	
6,084	28.6%	mobile	
2,322	10.9%	tablet	
21,244	100.0%	total	
ga:sessions			
16,014	59.7%	desktop	
7,644	28.5%	mobile	
3,159	11.8%	tablet	
26,817	100.0%	total	

To produce HTML, run:

```
clan report -f html analytics.html
```

3.1.2 Generating a JSON report

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

```
clan report -f json analytics.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 -f json --start-date 2014-05-1 --prefix /tshirt/ analytics.json
```

You can also convert an existing JSON report to text, like so:

```
clan report -d analytics.json analytics.txt
```

3.1.3 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.txt
```

This will write a report documenting the absolute and percentage point differences. Here is an example of the output:

Comparing report A run 2014-06-10 with:

```
property-id: 53470309
start-date: 2014-06-01
ndays: 2
prefix: /commencement/
```

With report B run 2014-06-10 with:

```
property-id: 53470309
start-date: 2014-06-01
ndays: 2
prefix: /tshirt/
```

Totals

ga:sessions	-12,280	-91.8%	-	total
ga:pageviews	-39,514	-96.3%	-	total
ga:users	-10,441	-91.9%	-	total
ga:uniquePageviews	-27,327	-96.2%	-	total

Totals by device category

ga:sessions	-3,832	-96.6%	-17.3	mobile
	-12,280	-91.8%	-	total
	-1,470	-92.9%	-1.5	tablet

	-6,978	-89.2%	18.8	desktop
ga:pageviews				
	-7,548	-97.8%	-7.5	mobile
	-39,514	-96.3%	-	total
	-4,608	-97.2%	-2.8	tablet
	-27,358	-95.8%	10.3	desktop
ga:users				
	-3,321	-97.1%	-19.4	mobile
	-10,441	-91.9%	-	total
	-1,204	-92.9%	-1.4	tablet
	-5,916	-89.0%	20.8	desktop
ga:uniquePageviews				
	-6,025	-97.8%	-9.1	mobile
	-27,327	-96.2%	-	total
	-3,589	-97.0%	-2.7	tablet
	-17,713	-95.5%	11.8	desktop

The values in the report columns are:

- Absolute difference
- Percent change
- Change in percentage points

3.1.4 Generating a JSON diff

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

```
clan diff -f json a.json b.json diff.json
```

3.2 Configuration

3.2.1 Configuring with YAML

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

By default clan will look for a YAML file called `clan.yml`. This can be configured using the `-c` command line flag. The basic structure of this file is:

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

# A list of queries to execute
queries:

  # Individual query configuration
  - name: Totals
    metrics:
      - "ga:pageviews"
      - "ga:uniquePageviews"
      - "ga:users"
      - "ga:sessions"
```

3.2.2 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

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.

3.2.3 Per-query configuration

Individual queries support the following properties.

name

A description of the query. Will be used as a display name when rendering a text report.

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.

3.3 Common queries

3.3.1 Total pageviews, uniques, users, etc.

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

3.3.2 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"
```

3.3.3 Browser share

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

3.3.4 Most viewed pages

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

3.3.5 Top sources (referrers)

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

3.3.6 Top social sources

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

3.3.7 Page load

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

3.3.8 Time on site

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

3.3.9 Custom event count

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

3.3.10 Custom event value

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

Authors

- Christopher Groskopf

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

6.1 0.1.3

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

6.2 0.1.2

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

6.3 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`.

6.4 0.1.0

- Initial version.

Indices and tables

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