Google Analytics Relay Documentation

Release 0.2.0

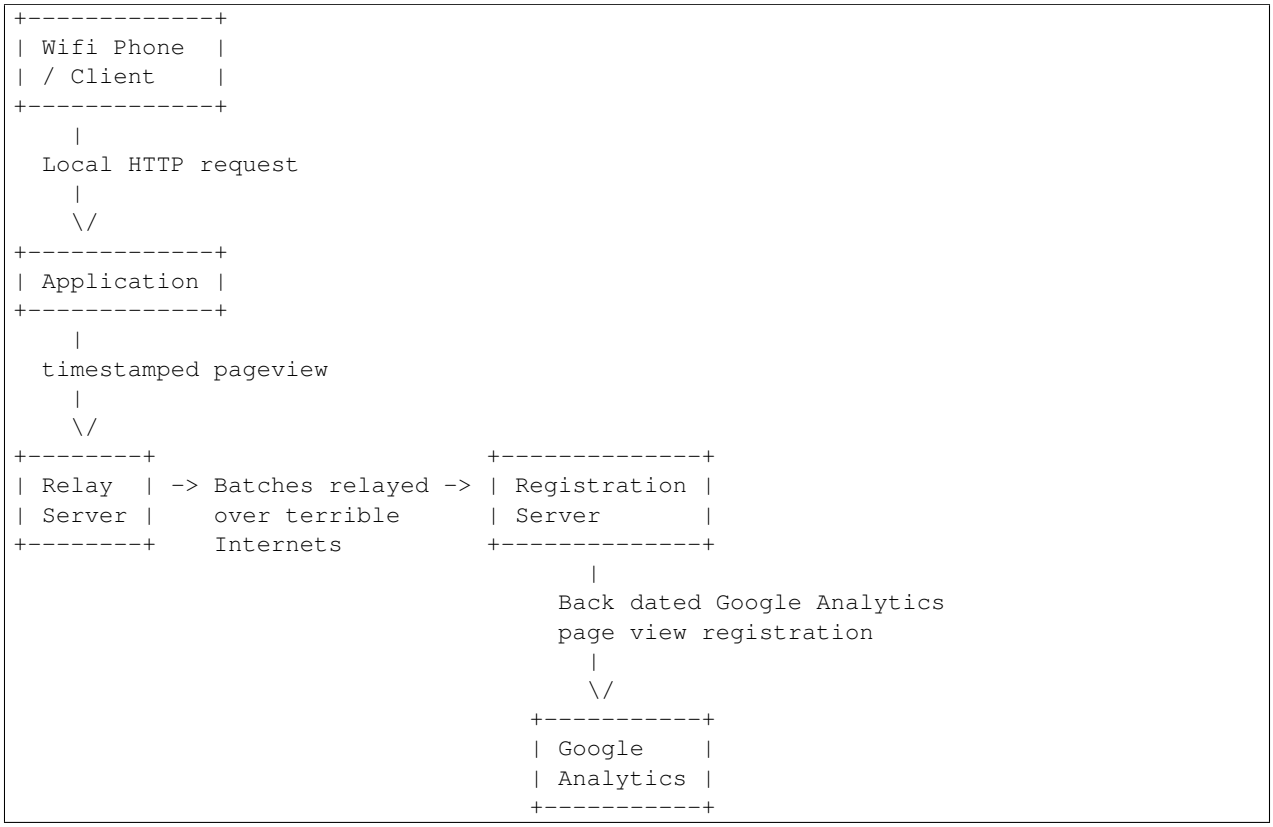
Praekelt Foundation

October 20, 2015

Contents

Capture pageviews in a somewhat offline capacity. Store them temporarily and relay them as batches to an upstream server (when connectivity is available). The upstream server then registers these page views with Google Analytics.

This works using a embedding a 1x1 pixel GIF image:



Note: Events are submitted to Google Analytics with a `queue time` parameter set. Google Analytics gives no guarantees about events that are submitted with a queue time of more than 4 hours ago.

CHAPTER 1

Proof

CHAPTER 2

Docker

```
$ docker run -it sdehaan/garelay [ tracker | server ]
```

CHAPTER 3

Embedding Directly

```
<img src="http://garelay/tracker-<GA-TRACKING-CODE>.gif?dp=/mycurrent/page.html">
```

GARelay will automatically include the following parameters:

uip The registered REMOTE_ADDR

dr The HTTP Referer

ul The Accept Language

Any of the Google Analytics tracking parameters can be passed along to the GIF. If you use Javascript to generate the `<img/>` tag then you can also include dynamic values such as device screen size.

CHAPTER 4

Proxying via Nginx

```
<img src="tracker-<GA-TRACKING-CODE>.gif">
```

Setting up Nginx to proxy anything matching `tracker-(?P<tracking_id>[A-Za-z0-9\-]+)\.gif` will result in the automatic inclusion of the Document Path parameter since the image is loaded relative to the current path:

If you URL is `http://www.example.com/mypage/hello/` and you embed the img tag there it will result in a `dp` (document path) value of `mypage/hello/` since the pixel is retrieved from the URL:

```
http://garelay/mypage/hello/tracker-<GA-TRACKING-CODE>.gif
```

Which sets the `dp` parameter.

CHAPTER 5

Overriding Default Values

Any of the values specified as extra querystring parameters will override the defaults.

CHAPTER 6

Pip Installation

```
$ virtualenv ve
$ source ve/bin/activate
(ve)$ pip install garelay
```

Run the Django registration server:

```
(ve)$ django-admin runserver --settings=garelay.server.settings
```

Run the Google Analytics tracker server:

```
(ve)$ django-admin runserver --settings=garelay.tracker.production
```

If you don't want to have to rely on Celery running you can schedule the tasks to be run via `cron` as well from the command line.

To relay the pageview events from the relay server to the registration server:

```
(ve)$ GARELAY_SERVER=http://www.example.com/ django-admin \
  --settings=garelay.tracker.settings \
  relay_events
```

To register the relayed pageview events at Google Analytics:

```
(ve)$ django-admin \
  --settings=garelay.server.settings \
  register_events
```