pypi-download-stats Documentation

Release 0.2.1

Jason Antman

Jul 06, 2018

Contents


build
passing

CHAPTER 1

Introduction

This package retrieves download statistics from Google BigQuery for one or more PyPI packages, caches them locally, and then generates download count badges as well as an HTML page of raw data and graphs (generated by bokeh). It's intended to be run on a schedule (i.e. daily) and have the results uploaded somewhere.

It would certainly be nice to make this into a real service (and some extension points for that have been included), but at the moment I have neither the time to dedicate to that, the money to cover some sort of hosting and bandwidth, nor the desire to handle how to architect this for over 85,000 projects as opposed to my few.

Hopefully stats like these will eventually end up in the official PyPI; see warehouse #699, #188 and #787 for reference on that work. For the time being, I want to (a) give myself a way to get simple download stats and badges like the old PyPI legacy (downloads per day, week and month) as well as (b) enable some higher-granularity analysis.

Note that this is a relatively heavy-weight solution; it has many dependencies and is really intended for people whose main need is to generate detailed historical graphs and download count badges for their projects. If your really just want to perform some ad-hoc queries, counts, or simple data analysis on the PyPI downloads dataset, a project like Ofek's pypinfo would be a simpler alternative.

Also note this package is *very* young; I wrote it as an evening/weekend project, hoping to only take a few days on it. Though writing this makes me want to bathe immediately, it has no tests. If people start using it, I'll change that.

For a live example of exactly how the output looks, you can see the download stats page for my awslimitchecker project, generated by a cronjob on my desktop, at: http://jantman-personal-public.s3-website-us-east-1.amazonaws.com/pypi-stats/awslimitchecker/index.html.

CHAPTER 2

Background

Sometime in February 2016, download stats stopped working on pypi.python.org. As I later learned, what we currently (August 2016) know as pypi is really the pypi-legacy codebase, and is far from a stable hands-off service. The small team of interpid souls who keep it running have their hands full simply keeping it online, while also working on its replacement, warehouse (which as of August 2016 is available online at https://pypi.io/). While the actual pypi.python.org web UI hasn't been switched over to the warehouse code yet (it's still under development), the current Warehouse service does provide full access to pypi. It's completely understandable that, given all this and the "life support" status of the legacy pypi codebase, download stats in a legacy codebase are their last concern.

However, current download statistics (actually the raw log information) since January 22, 2016 are available in a Google BigQuery public dataset and being updated in near-real-time. There may be download statistics functionality

CHAPTER 3

Requirements

- Python 2.7+ (currently tested with 2.7, 3.5, 3.6)
- Python VirtualEnv and `pip` (recommended installation method; your OS/distribution should have packages for these)

pypi-download-stats relies on bokeh to generate pretty SVG charts that work offline, and google-api-python-client for querying BigQuery. Each of those have additional dependencies.

CHAPTER 4

Installation

It's recommended that you install into a virtual environment (virtualenv / venv). See the virtualenv usage documentation for information on how to create a venv.

This isn't on pypi yet, ironically. Until it is:

```
$ pip install pypi-download-stats
```

CHAPTER 5

Configuration

You'll need Google Cloud credentials for a project that has the BigQuery API enabled. The recommended method is to generate system account credentials; download the JSON file for the credentials and export the path to it as the `GOOGLE_APPLICATION_CREDENTIALS` environment variable. The system account will need to be added as a Project Member.

CHAPTER 6

Usage

Run with `-h` for command-line help:

```
usage: pypi-download-stats [-h] [-V] [-v] [-Q | -G] [-o OUT_DIR]
                           [-p PROJECT_ID] [-c CACHE_DIR] [-B BACKFILL_DAYS]
                           [-P PROJECT | -U USER]

pypi-download-stats - Calculate detailed download stats and generate HTML and
badges for PyPI packages - <https://github.com/jantman/pypi-download-stats>

optional arguments:
  -h, --help            show this help message and exit
  -V, --version         show program's version number and exit
  -v, --verbose         verbose output. specify twice for debug-level output.
  -Q, --no-query        do not query; just generate output from cached data
  -G, --no-generate     do not generate output; just query data and cache
                        results
  -o OUT_DIR, --out-dir OUT_DIR
                        output directory (default: ./pypi-stats
  -p PROJECT_ID, --project-id PROJECT_ID
                        ProjectID for your Google Cloud user, if not using
                        service account credentials JSON file
  -c CACHE_DIR, --cache-dir CACHE_DIR
                        stats cache directory (default: ./pypi-stats-cache)
  -B BACKFILL_DAYS, --backfill-num-days BACKFILL_DAYS
                        number of days of historical data to backfill, if
                        missing (defaut: 7). Note this may incur BigQuery
                        charges. Set to -1 to backfill all available history.
  -P PROJECT, --project PROJECT
                        project name to query/generate stats for (can be
                        specified more than once; this will reduce query cost
                        for multiple projects)
  -U USER, --user USER  Run for all PyPI projects owned by the specifieduser.
```

To run queries and generate reports for PyPI projects "foo" and "bar", using a Google Cloud credentials JSON file at `foo.json`:

```
$ export GOOGLE_APPLICATION_CREDENTIALS=/foo.json
$ pypi-download-stats -P foo -P bar
```

To run queries but *not* generate reports for all PyPI projects owned by user "myname":

```
$ export GOOGLE_APPLICATION_CREDENTIALS=/foo.json
$ pypi-download-stats -G -U myname
```

To generate reports against cached query data for the project "foo":

```
$ export GOOGLE_APPLICATION_CREDENTIALS=/foo.json
$ pypi-download-stats -Q -P foo
```

To run nightly and upload results to a website-hosting S3 bucket, I use the following script via cron (note the paths are specific to my purpose; also note the two commands, as `s3cmd` does not seem to set the MIME type for the SVG images correctly):

```
#!/bin/bash -x

export GOOGLE_APPLICATION_CREDENTIALS=/home/jantman/.ssh/pypi-bigquery.json
cd /home/jantman/GIT/pypi-download-stats
bin/pypi-download-stats -vv -U jantman

# sync html files
~/venvs/foo/bin/s3cmd -r --delete-removed --stats --exclude='*.svg' sync pypi-stats␣
↪s3://jantman-personal-public/
# sync SVG and set mime-type, since s3cmd gets it wrong
~/venvs/foo/bin/s3cmd -r --delete-removed --stats --exclude='*.html' --mime-type=
↪'image/svg+xml' sync pypi-stats s3://jantman-personal-public/
```

6.1 Cost

At this point… I have no idea. Some of the download tables are 3+ GB per day. I imagine that backfilling historical data from the beginning of what's currently there (20160122) might incur quite a bit of data cost.

CHAPTER 7

Bugs and Feature Requests

Bug reports and feature requests are happily accepted via the GitHub Issue Tracker. Pull requests are welcome. Issues that don't have an accompanying pull request will be worked on as my time and priority allows.

CHAPTER 8

Development

To install for development:

1. Fork the pypi-download-stats repository on GitHub
2. Create a new branch off of master in your fork.

```
$ virtualenv pypi-download-stats
$ cd pypi-download-stats && source bin/activate
$ pip install -e git+git@github.com:YOURNAME/pypi-download-stats.git@BRANCHNAME
↪#egg=pypi-download-stats
$ cd src/pypi-download-stats
```

The git clone you're now in will probably be checked out to a specific commit, so you may want to `git checkout BRANCHNAME`.

8.1 Guidelines

- pep8 compliant with some exceptions (see pytest.ini)

8.2 Testing

There isn't any right now. I'm bad. If people actually start using this, I'll refactor and add tests, but for now this started as a one-night project.

8.3 Release Checklist

1. Open an issue for the release; cut a branch off master for that issue.
2. Confirm that there are CHANGES.rst entries for all major changes.

3. Ensure that Travis tests passing in all environments.
4. Ensure that test coverage is no less than the last release (ideally, 100%).
5. Increment the version number in pypi-download-stats/version.py and add version and release date to CHANGES.rst, then push to GitHub.
6. Confirm that README.rst renders correctly on GitHub.
7. Upload package to testpypi:
   - Make sure your ~/.pypirc file is correct (a repo called `test` for https://testpypi.python.org/pypi)
   - `rm -Rf dist`
   - `python setup.py register -r https://testpypi.python.org/pypi`
   - `python setup.py sdist bdist_wheel`
   - `twine upload -r test dist/*`
   - Check that the README renders at https://testpypi.python.org/pypi/pypi-download-stats
8. Create a pull request for the release to be merged into master. Upon successful Travis build, merge it.
9. Tag the release in Git, push tag to GitHub:
   - tag the release. for now the message is quite simple: `git tag -a X.Y.Z -m 'X.Y.Z released YYYY-MM-DD'`
   - push the tag to GitHub: `git push origin X.Y.Z`
11. Upload package to live pypi:
    - `twine upload dist/*`
10. make sure any GH issues fixed in the release were closed.

CHAPTER 9

Contents

9.1 Example Output:

For a live example of exactly how the output looks, you can see the download stats page for my awslimitchecker project, generated by a cronjob on my desktop, at: http://jantman-personal-public.s3-website-us-east-1.amazonaws.com/pypi-stats/awslimitchecker/index.html.

9.2 pypi_download_stats

9.2.1 pypi_download_stats package

9.2.1.1 Submodules

pypi_download_stats.dataquery module

pypi_download_stats.diskdatacache module

class pypi_download_stats.diskdatacache.**DiskDataCache**(*cache_path*)
Bases: object

_path_for_file(*project_name*, *date*)
Generate the path on disk for a specified project and date.

Parameters

- **project_name** – the PyPI project name for the data
- **date** (*datetime.datetime*) – the date for the data

Returns path for where to store this data on disk

Return type str

get (*project*, *date*)
Get the cache data for a specified project for the specified date. Returns None if the data cannot be found in the cache.

Parameters

- **project** (*str*) – PyPi project name to get data for
- **date** (*datetime.datetime*) – date to get data for

Returns dict of per-date data for project

Return type dict or None

get_dates_for_project (*project*)
Return a list of the dates we have in cache for the specified project, sorted in ascending date order.

Parameters **project** (*str*) – project name

Returns list of datetime.datetime objects

Return type datetime.datetime

set (*project*, *date*, *data*, *data_ts*)
Set the cache data for a specified project for the specified date.

Parameters

- **project** (*str*) – project name to set data for
- **date** (*datetime.datetime*) – date to set data for
- **data** (*dict*) – data to cache
- **data_ts** (*int*) – maximum timestamp in the BigQuery data table

pypi_download_stats.graphs module

pypi_download_stats.outputgenerator module

pypi_download_stats.projectstats module

pypi_download_stats.runner module

pypi_download_stats.version module

9.3 Changelog

9.3.1 Unreleased Changes

- Stop testing and supporting py33; start testing py36.
- Stop testing and supporting py34, as we rely on pandas that doesn't support 3.4.
- Many fixes for pep8/flakes and docs build.

9.3.2 0.2.1 (2016-09-18)

- Fix example cron S3 upload script in README.

9.3.3 0.2.0 (2016-09-18)

- Fix packaging bug where HTML templates weren't included in package (making this entire tool pretty useless)
- Fix unicode output error.

9.3.4 0.1.0 (2016-08-28)

- Initial release

CHAPTER 10

Indices and tables

- genindex
- modindex
- search

10.1 License

pypi-download-stats is licensed under the GNU Affero General Public License, version 3 or later. This shouldn't be much of a concern to most people.

Python Module Index

p

Index

Symbols

D

G

P

S