

Pseudo API for Google Trends

 [View license](#)


☆ **2k** stars  **581** forks

☆ Star

 Notifications

<> **Code**


! Issues 71

 Pull requests 13

▶ Actions

 Projects

 Security

 master ▼

Go to file



emlazzarin Merge pull request [#424](#) from NicolasRisi/pr/retries ...

on 2 Oct 2020

 **518**

[View code](#)

README.md

pytrends

Introduction

Unofficial API for Google Trends

Allows simple interface for automating downloading of reports from Google Trends. Only good until Google changes their backend again :-P. When that happens feel free to contribute!

Looking for maintainers!

Table of contents

- [Installation](#)
- [API](#)
 - [API Methods](#)

- [Common API parameters](#)
 - [Interest Over Time](#)
 - [Historical Hourly Interest](#)
 - [Interest by Region](#)
 - [Related Topics](#)
 - [Related Queries](#)
 - [Trending Searches](#)
 - [Top Charts](#)
 - [Suggestions](#)
- [Caveats](#)
- [Credits](#)

Installation

```
pip install pytrends
```

Requirements

- Written for Python 3.3+
- Requires Requests, lxml, Pandas

[back to top](#)

API

Connect to Google

```
from pytrends.request import TrendReq

pytrends = TrendReq hl='en-US', tz=360)
```

or if you want to use proxies as you are blocked due to Google rate limit:

```
from pytrends.request import TrendReq

pytrends = TrendReq(hl='en-US', tz=360, timeout=(10,25), proxies=
```

```
['https://34.203.233.13:80'], retries=2, backoff_factor=0.1,
requests_args={'verify':False})
```

- `timeout(connect, read)`
 - See explanation on this on [requests docs](#)
- `tz`
 - Timezone Offset
 - For example US CST is '360' (note **NOT** -360, Google uses timezone this way...)
- `proxies`
 - https proxies Google passed ONLY
 - list ['https://34.203.233.13:80', 'https://35.201.123.31:880', ..., ...]
- `retries`
 - number of retries total/connect/read all represented by one scalar
- `backoff_factor`
 - A backoff factor to apply between attempts after the second try (most errors are resolved immediately by a second try without a delay). urllib3 will sleep for:
 $\{\text{backoff factor}\} * (2 ^ (\{\text{number of total retries}\} - 1))$ seconds. If the `backoff_factor` is 0.1, then `sleep()` will sleep for [0.0s, 0.2s, 0.4s, ...] between retries. It will never be longer than `Retry.BACKOFF_MAX`. By default, backoff is disabled (set to 0).
- `requests_args`
 - A dict with additional parameters to pass along to the underlying requests library, for example `verify=False` to ignore SSL errors

Note: the parameter `hl` specifies host language for accessing Google Trends. Note: only https proxies will work, and you need to add the port number after the proxy ip address

Build Payload

```
kw_list = ["Blockchain"]
pytrends.build_payload(kw_list, cat=0, timeframe='today 5-y', geo='',
gprop='')
```

Parameters

- `kw_list`
 - *Required*
 - Keywords to get data for

[back to top](#)

API Methods

The following API methods are available:

- [Interest Over Time](#): returns historical, indexed data for when the keyword was searched most as shown on Google Trends' Interest Over Time section.
- [Historical Hourly Interest](#): returns historical, indexed, hourly data for when the keyword was searched most as shown on Google Trends' Interest Over Time section. It sends multiple requests to Google, each retrieving one week of hourly data. It seems like this would be the only way to get historical, hourly data.
- [Interest by Region](#): returns data for where the keyword is most searched as shown on Google Trends' Interest by Region section.
- [Related Topics](#): returns data for the related keywords to a provided keyword shown on Google Trends' Related Topics section.
- [Related Queries](#): returns data for the related keywords to a provided keyword shown on Google Trends' Related Queries section.
- [Trending Searches](#): returns data for latest trending searches shown on Google Trends' Trending Searches section.
- [Top Charts](#): returns the data for a given topic shown in Google Trends' Top Charts section.
- [Suggestions](#): returns a list of additional suggested keywords that can be used to refine a trend search.

[back to top](#)

Common API parameters

Many API methods use the following:

- `kw_list`
 - keywords to get data for
 - Example `['Pizza']`
 - Up to five terms in a list: `['Pizza', 'Italian', 'Spaghetti', 'Breadsticks', 'Sausage']`
 - Advanced Keywords
 - When using Google Trends dashboard Google may provide suggested narrowed search terms.
 - For example "iron" will have a drop down of "Iron Chemical Element, Iron Cross, Iron Man, etc" .
 - Find the encoded topic by using the `get_suggestions()` function and choose the most relevant one for you.
 - For example:


```
https://www.google.com/trends/explore#q=%2Fm%2F025rw19&cmpt=q
```
 - "%2Fm%2F025rw19" is the topic "Iron Chemical Element" to use this with `pytrends`
 - You can also use `pytrends.suggestions()` to automate this.
- `cat`
 - Category to narrow results
 - Find available categories by inspecting the url when manually using Google Trends. The category starts after `cat=` and ends before the next `&` or view this [wiki page containing all available categories](#)
 - For example: `"https://www.google.com/trends/explore#q=pizza&cat=71"`
 - `'71'` is the category
 - Defaults to no category
- `geo`
 - Two letter country abbreviation
 - For example United States is `'us'`
 - Defaults to World
 - More detail available for States/Provinces by specifying additional abbreviations
 - For example: Alabama would be `'US-AL'`
 - For example: England would be `'GB-ENG'`

- tz
 - Timezone Offset (in minutes)
 - For more information of Timezone Offset, [view this wiki page containing about UCT offset](#)
 - For example US CST is '360'
- timeframe
 - Date to start from
 - Defaults to last 5yrs, 'today 5-y' .
 - Everything 'all'
 - Specific dates, 'YYYY-MM-DD YYYY-MM-DD' example '2016-12-14 2017-01-25'
 - Specific datetimes, 'YYYY-MM-DDTHH YYYY-MM-DDTHH' example '2017-02-06T10 2017-02-12T07'
 - Note Time component is based off UTC
 - Current Time Minus Time Pattern:
 - By Month: 'today #-m' where # is the number of months from that date to pull data for
 - For example: 'today 3-m' would get data from today to 3months ago
 - **NOTE** Google uses UTC date as 'today'
 - **Works for 1, 3, 12 months only!**
 - Daily: 'now #-d' where # is the number of days from that date to pull data for
 - For example: 'now 7-d' would get data from the last week
 - **Works for 1, 7 days only!**
 - Hourly: 'now #-H' where # is the number of hours from that date to pull data for
 - For example: 'now 1-H' would get data from the last hour
 - **Works for 1, 4 hours only!**
- gprop

- What Google property to filter to
- Example 'images'
- Defaults to web searches
- Can be images , news , youtube or froogle (for Google Shopping results)

[back to top](#)

Interest Over Time

```
pytrends.interest_over_time()
```

Returns pandas.DataFrame

[back to top](#)

Historical Hourly Interest

```
pytrends.get_historical_interest(kw_list, year_start=2018, month_start=1,
day_start=1, hour_start=0, year_end=2018, month_end=2, day_end=1,
hour_end=0, cat=0, geo='', gprop='', sleep=0)
```

Parameters

- kw_list
 - *Required*
 - list of keywords that you would like the historical data
- year_start, month_start, day_start, hour_start, year_end, month_end, day_end, hour_end
 - the time period for which you would like the historical data
- sleep
 - If you are rate-limited by Google, you should set this parameter to something (i.e. 60) to space off each API call.

Returns pandas.DataFrame

[back to top](#)

Interest by Region

```
pytrends.interest_by_region(resolution='COUNTRY', inc_low_vol=True, inc_geo_code=False)
```

Parameters

- resolution
 - 'CITY' returns city level data
 - 'COUNTRY' returns country level data
 - 'DMA' returns Metro level data
 - 'REGION' returns Region level data
- inc_low_vol
 - True/False (includes google trends data for low volume countries/regions as well)
- inc_geo_code
 - True/False (includes ISO codes of countries along with the names in the data)

Returns pandas.DataFrame

[back to top](#)

Related Topics

```
pytrends.related_topics()
```

Returns dictionary of pandas.DataFrames

[back to top](#)

Related Queries

```
pytrends.related_queries()
```

Returns dictionary of pandas.DataFrames

[back to top](#)

Trending Searches


```
pytrends.trending_searches(pn='united_states') # trending searches in real
time for United States
pytrends.trending_searches(pn='japan') # Japan
```

Returns pandas.DataFrame

[back to top](#)

Top Charts

```
pytrends.top_charts(date, hl='en-US', tz=300, geo='GLOBAL')
```

Parameters

- date
 - *Required*
 - YYYY integer
 - Example 2019 for the year 2019 Top Chart data
 - **Note** Google removed support for monthly queries (e.g. YYYY-MM)
 - **Note** Google does not return data for the current year

Returns pandas.DataFrame

[back to top](#)

Suggestions

```
pytrends.suggestions(keyword)
```

Parameters

- keyword
 - *Required*
 - keyword to get suggestions for

Returns dictionary

[back to top](#)

Categories

```
pytrends.categories()
```

Returns dictionary

[back to top](#)

Caveats

- This is not an official or supported API
- Google may change aggregation level for items with very large or very small search volume
- Rate Limit is not publicly known, let me know if you have a consistent estimate
 - One user reports that 1,400 sequential requests of a 4 hours timeframe got them to the limit. (Replicated on 2 networks)
 - It has been tested, and 60 seconds of sleep between requests (successful or not) is the correct amount once you reach the limit.
- For certain configurations the dependency lib certifi requires the environment variable REQUESTS_CA_BUNDLE to be explicitly set and exported. This variable must contain the path where the ca-certificates are saved or a SSLError: [SSL: CERTIFICATE_VERIFY_FAILED] error is given at runtime.

Credits

- Major JSON revision ideas taken from pat310's JavaScript library
 - <https://github.com/pat310/google-trends-api>
- Connecting to google code heavily based off Stack Overflow post
 - <http://stackoverflow.com/questions/6754709/logging-in-to-google-using-python>
- With some ideas pulled from Matt Reid's Google Trends API
 - <https://bitbucket.org/mattreid9956/google-trend-api/overview>

Releases

No releases published

NO RELEASES PUBLISHED

Packages

No packages published

Used by 801



Contributors 58



+ 47 contributors

Languages

● Python 100.0%