Flask-Track-Usage Documentation

Release 1.1.0

Steve Milner

May 30, 2018

Contents

Basic metrics tracking for your Flask application. This focuses more on ip addresses/locations rather than tracking specific users pathing through an application. No extra cookies or javascript is used for usage tracking.

- Simple. It's a Flask extension.
- Supports either include or exempt for views.
- Provides lite abstraction for data retrieval.
- Optional freegeoip.net integration including custom freegeoip installs.
- Multiple storage options.

CHAPTER 1

Installation

Warning: 1.1.x releases are not 100% backwards compatible with the 1.x.x nor 0.0.x series of releases.

1.1 Requirements

- Flask: http://flask.pocoo.org/
- A storage object to save the metrics data with

1.2 Via pip

```
$ pip install Flask-Track-Usage
```

1.3 Via source

```
$ python setup.py install
```

CHAPTER 2

Usage

```python
# Create the Flask 'app'
from flask import Flask
app = Flask(__name__)

# Set the configuration items manually for the example
app.config['TRACK_USAGE_USE_FREEGEOIP'] = False
# You can use a different instance of freegeoip like so
# app.config['TRACK_USAGE_FREEGEOIP_ENDPOINT'] = 'https://example.org/api/'
app.config['TRACK_USAGE_INCLUDE_OR_EXCLUDE_VIEWS'] = 'include'

# We will just print out the data for the example
from flask.ext.track_usage import TrackUsage
from flask.ext.track_usage.storage.printer import PrintStorage

# Make an instance of the extension
t = TrackUsage(app, PrintStorage())

# Make an instance of the extension
t = TrackUsage(app, storage)

# Include the view in the metrics
@t.include
@app.route('/')
def index():
    return "Hello"

# Run the application!
app.run(debug=True)
```

CHAPTER 3

Blueprint Support

Blueprints can be included or excluded from Flask-TrackUsage in their entirety.

3.1 Include

```
# ...
app.config['TRACK_USAGE_INCLUDE_OR_EXCLUDE_VIEWS'] = 'include'

# Make an instance of the extension
t = TrackUsage(app, PrintStorage())

from my_blueprints import a_bluprint

# Now ALL of a_blueprint's views will be in the include list
t.include_blueprint(a_blueprint)
```

3.2 Exclude

```
# ...
app.config['TRACK_USAGE_INCLUDE_OR_EXCLUDE_VIEWS'] = 'exclude'

# Make an instance of the extension
t = TrackUsage(app, PrintStorage())

from my_blueprints import a_bluprint

# Now ALL of different_blueprints will be in the exclude list
t.exclude_blueprint(a_blueprint)
```

CHAPTER 4

Configuration

4.1 TRACK_USAGE_USE_FREEGEOIP

Values: True, False

Default: False

4.2 TRACK_USAGE_FREEGEOIP_ENDPOINT

Values: URL for RESTful JSON query

Default: "http://extreme-ip-lookup.com/json/{ip}"

If TRACK_USAGE_USE_FREEGEOIP is True, then this field must be set. Mark the location for the IP address with "{ip}". For example:

> "http://example.com/{ip}/?key=484848484abc321"

would resolve (with an IP of 1.2.3.4) to:

> "http://example.com/1.2.3.4/?key=484848484abc321"

If using SQLStorage, the returned JSON is converted to a string. You will likely want to pass a field list in the URL to avoid exceeding the 128 character limit of the field.

Turn FreeGeoIP integration on or off

4.3 TRACK_USAGE_INCLUDE_OR_EXCLUDE_VIEWS

Values: include, exclude

Default: exclude

If views should be included or excluded by default.

- When set to *exclude* each routed view must be explicitly included via decorator or blueprint include method. If a routed view is not included it will not be tracked.
- When set to *include* each routed view must be explicitly excluded via decorator or blueprint exclude method. If a routed view is not excluded it will be tracked.

CHAPTER 5

Storage

The following are built in, ready to use storage backends.

Note: Inputs for set_up should be passed in __init__ when creating a storage instance

5.1 printer.PrintStorage

Note: This storage backend is only for testing!

5.2 couchdb.CouchDBStorage

5.3 mongo.MongoPiggybackStorage

5.4 mongo.MongoStorage

5.5 redis_db.RedisStorage

5.6 sql.SQLStorage

Warning: This storage is not backwards compatible with sql.SQLStorage 1.0.x

CHAPTER 6

Retrieving Data

All storage backends, other than printer.PrintStorage, provide get_usage.

Results that are returned from all instances of get_usage should **always** look like this:

```
[
    {
            'url': str,
            'user_agent': {
                'browser': str,
                'language': str,
                'platform': str,
                'version': str,
            },
            'blueprint': str,
            'view_args': dict or None
            'status': int,
            'remote_addr': str,
            'xforwardedfor': str,
            'authorization': bool
            'ip_info': str or None,
            'path': str,
            'speed': float,
            'date': datetime,
    },
    {
        ....
    }
]
```

Changed in version 1.1.0: xforwardfor item added directly after remote_addr