
durandal Documentation

Release latest

Jul 24, 2018

Contents

1	Additional info	3
2	Requerimentss	5
3	Importing and using durandal	7
4	durandal private class functions	9

information needed in order to use durandal lib to digest and index nginx default logs format.

CHAPTER 1

Additional info

This was customized to work with <https://dotahotkeys.com> page format. In case you wish to customize for your nginx webserver, you can fork it or just PM me at contato.carmando@gmail.com.

ps.: It also has a iplocation.net wrapper in order to grab information on the ip address that accessed the site.

CHAPTER 2

Requerimentss

- Python 3.6
- PIP for python 3.6
- install `requirements.txt` file
- Internet access, in order to use `iplocation.net` wrapper

In order to install the requirements file, use the followings commands:

Windows Users: \$ `pip install -r requirements.txt`

Linux/macOS Users: \$ `pip3.6 install -r requirements.txt`

Importing and using durandal

Download `durandal.py` and `__init__.py` files and place it on the folder `site-packages/durandal/` of your system (create folder `durandal` in case it does not exist).

Simple usage of `durandal`:

```
from durandal import *
coordinator = durandal("/Users/carmando/Downloads/log/")
coordinator.dbfile = "/Users/carmando/PycharmProjects/Durandal/venv/output.db"
↪
coordinator.generateReport(onlyMatch="/search/")
```

`durandal()` will initialize `durandal`'s log path (keep in mind that the default `nginx` log will have the prefix name of `access.log`)

`dbfile` is the output `sqlite3` database name/path that will be created after `durandal` digest all the log information

`generateReport(onlyMatch="": str)` is the function used to generate the `db` file and start to ingest all the logs on the previously initiated path. it takes `onlyMatch` as arguments, meaning that it will only parse lines that have the selected string argument.

CHAPTER 4

durandal private class functions

`outputDB(self, *args)` function used to generate the new db (file and its tables schema) in case it does not exist, and populate the db with the acquired information (it receives the data to push to the table args **args*).

`getIpInfo(self, ip:str)` wrapper built to consult iplocation.net website with the provided ip address. Will return a list of [ip, country, region city].

`splitDateAndHour(self, string:str)` function used to split nginx dateHour string into a list of date and hour. Will return a list of [date, hour]