
django-proxylist-for-grab Documentation

Release 0.4

gotlium

May 27, 2017

Contents

1	Introduction	3
1.1	Checking proxies	3
2	Dependencies	5
2.1	Packages	5
2.2	Backend	5
3	Installation	7
3.1	Installing the package	7
3.2	Configuration	7
3.3	Database creation	7
4	Use case	9
5	Command reference	11
5.1	update_proxies	11
5.2	check_proxies	11
5.3	grab_proxies	11
5.4	clean_proxies	11
6	Advanced configuration	13
7	Indices and tables	15

django-proxylist-for-grab is a reusable app for maintain an up-to-date (through checking) list of proxy servers.

Contents:

Checking proxies

Proxy checking is the core of *django-proxylist-for-grab*, in order to make it you need three things:

- A checker: It shots a request throught the proxy.
- A proxy: Of course ...
- A mirror: It's a special page that reflects the checker request and allow us to see how an third page would see us.

django-proxylist-for-grab depends on:

Packages

- django-celery
- django-countries
- pygeoip
- grub

Backend

Cache

The checking machinery uses django's cache backend and the *default* cache but you can alter this behaviour changing the **PROXYLIST_CACHE** variable.

Database

django-proxylist-for-grab does not depends on any database backend by itself, but if you have a big list of proxies and you want to check it at sorts intervals you should avoid *SQLite*.

Installing the package

django-proxylist-for-grab can be easily installed using pip:

```
$ pip install django-proxylist-for-grab
```

Configuration

After that you need to include *django-proxylist-for-grab* into your *INSTALLED_APPS* list of your django settings file.

```
INSTALLED_APPS = (  
    ...  
    'proxylist',  
    ...  
)
```

django-proxylist-for-grab has a list of variables that you can configure through django's settings file. You can see the entire list at Advanced Configuration.

Database creation

You have two choices here:

Using south

We encourage you using *south* for your database migrations. If you already use it you can migrate *django-proxylist-for-grab*:

```
$ python manage.py migrate proxylist
```

Using syncdb

If you don't want to use *south* you can make a plain *syncdb*:

```
$ python manage.py syncdb
```

CHAPTER 4

Use case

Command reference

update_proxies

Add new proxies from a file.

```
$ python manage.py update_proxies [file1] <file2> <...>
```

check_proxies

Check proxies availability and anonymity.

```
$ python manage.py check_proxies
```

grab_proxies

Search proxy list on internet

```
$ python manage.py grab_proxies
```

clean_proxies

Remove broken proxies

```
$ python manage.py clean_proxies
```

Advanced configuration

PROXYLIST_CACHE_TIMEOUT

Maximum number of seconds to maintain a lock at the cache framework.

Default: 0

PROXYLIST_CONNECTION_TIMEOUT

Number of seconds to wait for a connection to open, before canceling the attempt and generate an error.

Default: 30

PROXYLIST_ERROR_DELAY

Number of seconds to add to each check if the last one produced an error.

Default: 300

PROXYLIST_GEOIP_PATH

Path to GeoIP data file.

Default: /usr/share/GeoIP/GeoIP.dat

PROXYLIST_MAX_CHECK_INTERVAL

Maximum number of seconds to the next check if the last one was successful.

Default: 900

PROXYLIST_MIN_CHECK_INTERVAL

Minimum number of seconds to the next check if the last one was successful.

Default: 300

PROXYLIST_OUTIP_INTERVAL

Number of seconds between outbound IP checking (per worker). If you have a fixed IP address you can set this value to 0 (infinity).

Default: 300

PROXYLIST_USER_AGENT

User-Agent for requests.

Default: Django-ProxyList 1.0.0

CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`