scrapy-crawlera Documentation

Release 1.5.0

Scrapinghub

May 12, 2021

CONFIGURATION

1 Configuration **3**

2 How to use it **5**
2.1 Settings 5

3 All the rest **9**
3.1 Changes 9

scrapy-crawlera is a Scrapy Downloader Middleware to interact with Crawlera automatically.

CHAPTER

ONE

CONFIGURATION

- Add the Crawlera middleware including it into the `DOWNLOADER_MIDDLEWARES` in your `settings.py` file:

```
DOWNLOADER_MIDDLEWARES = {
    ...
    'scrapy_crawlera.CrawleraMiddleware': 610
}
```

- Then there are two ways to enable it
  - Through `settings.py`:

```
CRAWLERA_ENABLED = True
CRAWLERA_APIKEY = 'apikey'
```

  - Through spider attributes:

```
class MySpider:
    crawlera_enabled = True
    crawlera_apikey = 'apikey'
```

Hint: You can also use *CRAWLERA_USER* and *CRAWLERA_PASS* instead of *CRAWLERA_APIKEY*.

- (optional) If you are not using the default Crawlera proxy (`http://proxy.crawlera.com:8010`), for example if you have a dedicated or private instance, make sure to also set `CRAWLERA_URL` in `settings.py`, e.g.:

```
CRAWLERA_URL = 'http://myinstance.crawlera.com:8010'
```

CHAPTER

TWO

HOW TO USE IT

2.1 Settings

This Middleware adds some settings to configure how to work with Crawlera.

2.1.1 CRAWLERA_APIKEY

Default: `None`

Unique Crawlera API Key provided for authentication.

2.1.2 CRAWLERA_USER

deprecated

Default: `None`

Old authentication user for the Crawlera service. It works with *CRAWLERA_PASS*.

2.1.3 CRAWLERA_PASS

deprecated

Default: `None`

Old authentication password for the Crawlera service: It works with *CRAWLERA_USER*.

2.1.4 CRAWLERA_URL

Default: `'http://proxy.crawlera.com:8010'`

Crawlera instance url, it varies depending on adquiring a private or dedicated instance. If Crawlera didn't provide you with a private instance url, you don't need to specify it.

2.1.5 CRAWLERA_MAXBANS

Default: `400`

Number of consecutive bans from Crawlera necessary to stop the spider.

2.1.6 CRAWLERA_DOWNLOAD_TIMEOUT

Default: `190`

Timeout for processing Crawlera requests. It overrides Scrapy's `DOWNLOAD_TIMEOUT`.

2.1.7 CRAWLERA_PRESERVE_DELAY

Default: `False`

If `False` Sets Scrapy's `DOWNLOAD_DELAY` to `0`, making the spider to crawl faster. If set to `True`, it will respect the provided `DOWNLOAD_DELAY` from Scrapy.

2.1.8 CRAWLERA_DEFAULT_HEADERS

Default: `{}`

Default headers added only to crawlera requests. Headers defined on `DEFAULT_REQUEST_HEADERS` will take precedence as long as the `CrawleraMiddleware` is placed after the `DefaultHeadersMiddleware`. Headers set on the requests have precedence over the two settings.

- This is the default behavior, `DefaultHeadersMiddleware` default priority is `400` and we recommend `CrawleraMiddleware` priority to be `610`

2.1.9 CRAWLERA_BACKOFF_STEP

Default: `15`

Step size used for calculating exponential backoff according to the formula: `random.uniform(0, min(max, step * 2 ** attempt))`.

2.1.10 CRAWLERA_BACKOFF_MAX

Default: `180`

Max value for exponential backoff as showed in the formula above.

Settings All configurable Scrapy Settings added by the Middleware.

With the middleware, the usage of crawlera is automatic, every request will go through crawlera without nothing to worry about. If you want to *disable* crawlera on a specific Request, you can do so by updating *meta* with *dont_proxy=True*:

```
scrapy.Request(
    'http://example.com',
    meta={
        'dont_proxy': True,
```

(continues on next page)

(continued from previous page)

```
        ...
    },
)
```

Remember that you are now making requests to Crawlera, and the Crawlera service will be the one actually making the requests to the different sites.

If you need to specify special Crawlera Headers, just apply them as normal Scrapy Headers.

Here we have an example of specifying a Crawlera header into a Scrapy request:

```
scrapy.Request(
    'http://example.com',
    headers={
        'X-Crawlera-Max-Retries': 1,
        ...
    },
)
```

Remember that you could also set which headers to use by default by all requests with DEFAULT_REQUEST_HEADERS

Note: Crawlera headers are removed from requests when the middleware is activated but Crawlera is disabled. For example, if you accidentally disable Crawlera via `crawlera_enabled = False` but keep sending `X-Crawlera-*` headers in your requests, those will be removed from the request headers.

This Middleware also adds some configurable Scrapy Settings, check *the complete list here*.

CHAPTER

THREE

ALL THE REST

3.1 Changes

3.1.1 v1.5.0 (2019-01-23)

- Correctly check for bans in crawlera (Jobs will not get banned on non ban 503's).
- Exponential backoff when crawlera doesn't have proxies available.
- Fix `dont_proxy=False` header disabling crawlera when it is enabled.

3.1.2 v1.4.0 (2018-09-20)

- Remove X-Crawlera-* headers when Crawlera is disabled.
- Introduction of DEFAULT_CRAWLERA_HEADERS settings.

3.1.3 v1.3.0 (2018-01-10)

- Use CONNECT method to contact Crawlera proxy.

3.1.4 v1.2.4 (2017-07-04)

- Trigger PYPI deployments after changes made to TOXENV in v1.2.3

3.1.5 v1.2.3 (2017-06-29)

- Multiple documentation fixes
- Test scrapy-crawlera on combinations of software used by scrapinghub stacks

3.1.6 v1.2.2 (2017-01-19)

- Fix Crawlera error stats key in Python 3.
- Add support for Python 3.6.

3.1.7 v1.2.1 (2016-10-17)

- Fix release date in README.

3.1.8 v1.2.0 (2016-10-17)

- Recommend middleware order to be `610` to run before `RedirectMiddleware`.
- Change default download timeout to 190s or 3 minutes 10 seconds (instead of 1800s or 30 minutes).
- Test and advertize Python 3 compatiblity.
- New `crawlera/request` and `crawlera/request/method/*` stats counts.
- Clear Scrapy DNS cache for proxy URL in case of connection errors.
- Distribute plugin as universal wheel.

Changes See what has changed in recent scrapy-crawlera versions.