
scrapy-zyte-smartproxy Documentation

Release 2.4.1

Zyte

May 04, 2026

CONTENTS

1	Headers	1
2	Stats	3
3	Settings	5
3.1	ZYTE_SMARTPROXY_APIKEY	5
3.2	ZYTE_SMARTPROXY_URL	5
3.3	ZYTE_SMARTPROXY_MAXBANS	5
3.4	ZYTE_SMARTPROXY_DOWNLOAD_TIMEOUT	5
3.5	ZYTE_SMARTPROXY_PRESERVE_DELAY	5
3.6	ZYTE_SMARTPROXY_DEFAULT_HEADERS	6
3.7	ZYTE_SMARTPROXY_BACKOFF_STEP	6
3.8	ZYTE_SMARTPROXY_BACKOFF_MAX	6
3.9	ZYTE_SMARTPROXY_FORCE_ENABLE_ON_HTTP_CODES	6
3.10	ZYTE_SMARTPROXY_KEEP_HEADERS	6
4	Changes	7
4.1	v2.4.1 (2025-03-24)	7
4.2	v2.4.0 (2024-12-30)	7
4.3	v2.3.5 (2024-08-05)	7
4.4	v2.3.4 (2024-05-09)	7
4.5	v2.3.3 (2024-02-22)	7
4.6	v2.3.2 (2024-02-14)	7
4.7	v2.3.1 (2023-11-20)	8
4.8	v2.3.0 (2023-10-20)	8
4.9	v2.2.0 (2022-08-05)	8
4.10	v2.1.0 (2021-06-16)	8
4.11	v2.0.0 (2021-05-12)	8
4.12	v1.7.2 (2020-12-01)	8
4.13	v1.7.1 (2020-10-22)	9
4.14	v1.7.0 (2020-04-01)	9
4.15	v1.6.0 (2019-05-27)	9
4.16	v1.5.1 (2019-05-21)	9
4.17	v1.5.0 (2019-01-23)	9
4.18	v1.4.0 (2018-09-20)	9
4.19	v1.3.0 (2018-01-10)	9
4.20	v1.2.4 (2017-07-04)	10
4.21	v1.2.3 (2017-06-29)	10
4.22	v1.2.2 (2017-01-19)	10
4.23	v1.2.1 (2016-10-17)	10

4.24 v1.2.0 (2016-10-17)	10
5 Configuration	11
6 Usage	13

HEADERS

The Zyte proxy services that you can use with this downloader middleware each support a different set of HTTP request and response headers that give you access to additional features. You can find more information about those headers in the documentation of each service, [Zyte API's](#) and [Zyte Smart Proxy Manager's](#).

If you try to use a header for one service while using the other service, this downloader middleware will try to translate your header into the right header for the target service and, regardless of whether or not translation was done, the original header will be dropped.

Also, response headers that can be translated will be always translated, without dropping the original header, so code expecting a response header from one service can work even if a different service was used.

Translation is supported for the following headers:

Zyte API	Zyte Smart Proxy Manager
Zyte-Device	X-Crawlera-Profile
Zyte-Error-Type	X-Crawlera-Error
Zyte-Geolocation	X-Crawlera-Region
Zyte-JobId	X-Crawlera-JobId
Zyte-Override-Headers	X-Crawlera-Profile-Pass

Also, if a request is not being proxied and includes a header for any of these services, it will be dropped, to prevent leaking data to external websites. This downloader middleware assumes that a header prefixed with Zyte- is a Zyte API header, and that a header prefixed with X-Crawlera- is a Zyte Smart Proxy Manager header, even if they are not known headers otherwise.

When dropping a header, be it as part of header translation or to avoid leaking data, a warning message with details will be logged.

STATS

This Scrapy plugin tracks some stats.

Stats for the [proxy mode](#) of [Zyte API](#) and stats for [Zyte Smart Proxy Manager](#) (formerly Crawlera) have a different prefix, `zyte_api_proxy` and `zyte_smartproxy` respectively.

SETTINGS

This Scrapy downloader middleware adds some settings to configure how to work with your Zyte proxy service.

3.1 ZYTE_SMARTPROXY_APIKEY

Default: None

Default API key for your Zyte proxy service.

Note that Zyte API and Zyte Smart Proxy Manager have different API keys.

You can *override this value on specific requests*.

3.2 ZYTE_SMARTPROXY_URL

Default: 'http://proxy.zyte.com:8011'

Default endpoint for your Zyte proxy service.

For guidelines on setting a value, see the *initial configuration instructions*.

You can *override this value on specific requests*.

3.3 ZYTE_SMARTPROXY_MAXBANS

Default: 400

Number of consecutive bans necessary to stop the spider.

3.4 ZYTE_SMARTPROXY_DOWNLOAD_TIMEOUT

Default: 190

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

3.5 ZYTE_SMARTPROXY_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.

3.6 ZYTE_SMARTPROXY_DEFAULT_HEADERS

Default: {}

Default headers added only to proxied requests. Headers defined on `DEFAULT_REQUEST_HEADERS` will take precedence as long as the `ZyteSmartProxyMiddleware` 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 `ZyteSmartProxyMiddleware` priority to be 610.

3.7 ZYTE_SMARTPROXY_BACKOFF_STEP

Default: 15

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

3.8 ZYTE_SMARTPROXY_BACKOFF_MAX

Default: 180

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

3.9 ZYTE_SMARTPROXY_FORCE_ENABLE_ON_HTTP_CODES

Default: []

List of HTTP response status codes that warrant enabling your Zyte proxy service for the corresponding domain.

When a response with one of these HTTP status codes is received after an unproxied request, the request is retried with your Zyte proxy service, and any new request to the same domain is also proxied.

3.10 ZYTE_SMARTPROXY_KEEP_HEADERS

Default: False

If True, header dropping and translation is disabled.

CHANGES

4.1 v2.4.1 (2025-03-24)

Stop expecting a `Zyte-Error` header in responses from `Zyte API proxy mode`, it is named `Zyte-Error-Type`.

4.2 v2.4.0 (2024-12-30)

Removed official support for Python 3.4, 3.5, 3.6, 3.7 and 3.8. Added official Python 3.13 support.

Backward-compatible `X-Crawlera`-prefixed headers are no longer translated into their matching `Zyte API proxy mode` headers, `Zyte API` now handles their translation on the server side.

Added a new `ZYTE_SMARTPROXY_KEEP_HEADERS` setting that allows disabling header dropping and translation.

4.3 v2.3.5 (2024-08-05)

Ban and throttling responses from `Zyte API proxy mode` are now handled in line with matching responses from `Zyte Smart Proxy Manager`.

4.4 v2.3.4 (2024-05-09)

`Zyte API proxy mode` now has its own stat prefix.

Some user-facing messages mentioning only `Zyte Smart Proxy Manager` have also been updated to reflect the fact that `scrapy-zyte-smartproxy` also supports `Zyte API proxy mode`.

4.5 v2.3.3 (2024-02-22)

Fix response handling for `Zyte API proxy mode`. Before, a single connection issue during a request would add a 90 second delay between requests until the end of the crawl, instead of removing the delay after the first successful response.

4.6 v2.3.2 (2024-02-14)

Detect scenarios where the `proxy Request.meta` key has probably been accidentally copied from an earlier response, warn about it, and fix the value.

The `Zyte-Client` header is again sent when using `Zyte API proxy mode`, now that `Zyte API` supports it.

4.7 v2.3.1 (2023-11-20)

Fixed Zyte API proxy mode support by removing the mapping of unsupported headers Zyte-Client and Zyte-No-Bancheck.

4.8 v2.3.0 (2023-10-20)

Added support for the upcoming proxy mode of Zyte API.

Added a BSD-3-Clause license file.

4.9 v2.2.0 (2022-08-05)

Added support for Scrapy 2.6.2 and later.

Scrapy 1.4 became the minimum supported Scrapy version.

4.10 v2.1.0 (2021-06-16)

- Use a custom logger instead of the root one

4.11 v2.0.0 (2021-05-12)

Following the upstream rebranding of Crawlera as Zyte Smart Proxy Manager, scrapy-crawlera has been renamed as scrapy-zyte-smartproxy, with the following backward-incompatible changes:

- The repository name and Python Package Index (PyPI) name are now scrapy-zyte-smartproxy.
- Setting prefixes have switched from CRAWLERA_ to ZYTE_SMARTPROXY_.
- Spider attribute prefixes and request meta key prefixes have switched from crawlera_ to zyte_smartproxy_.
- scrapy_crawlera is now scrapy_zyte_smartproxy.
- CrawleraMiddleware is now ZyteSmartProxyMiddleware, and its default url is now http://proxy.zyte.com:8011.
- Stat prefixes have switched from crawlera/ to zyte_smartproxy/.
- The online documentation is moving to <https://scrapy-zyte-smartproxy.readthedocs.io/>

Note

Zyte Smart Proxy Manager headers continue to use the X-Crawlera- prefix.

- In addition to that, the X-Crawlera-Client header is now automatically included in all requests.

4.12 v1.7.2 (2020-12-01)

- Use request.meta than response.meta in the middleware

4.13 v1.7.1 (2020-10-22)

- Consider Crawlera response if contains *X-Crawlera-Version* header
- Build the documentation in Travis CI and fail on documentation issues
- Update matrix of tests

4.14 v1.7.0 (2020-04-01)

- Added more stats to better understanding the internal states.
- Log warning when using *https://* protocol.
- Add default *http://* protocol in case of none provided, and log warning about it.
- Fix duplicated request when the response is not from crawlera, this was causing an infinite loop of retries when *dont_filter=True*.

4.15 v1.6.0 (2019-05-27)

- Enable crawlera on demand by setting `CRAWLERA_FORCE_ENABLE_ON_HTTP_CODES`

4.16 v1.5.1 (2019-05-21)

- Remove username and password from settings since it's removed from crawlera.
- Include affected spider in logs.
- Handle situations when crawlera is restarted and reply with 407's for a few minutes by retrying the requests with a exponential backoff system.

4.17 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.

4.18 v1.4.0 (2018-09-20)

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

4.19 v1.3.0 (2018-01-10)

- Use `CONNECT` method to contact Crawlera proxy.

4.20 v1.2.4 (2017-07-04)

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

4.21 v1.2.3 (2017-06-29)

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

4.22 v1.2.2 (2017-01-19)

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

4.23 v1.2.1 (2016-10-17)

- Fix release date in README.

4.24 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 compatibility.
- 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.

scrapy-zyte-smartproxy is a [Scrapy downloader middleware](#) to use one of Zyte's proxy services: either the [proxy mode](#) of [Zyte API](#) or [Zyte Smart Proxy Manager](#) (formerly [Crawlera](#)).

CONFIGURATION

1. Add the downloader middleware to your DOWNLOADER_MIDDLEWARES Scrapy setting:

Listing 1: settings.py

```
DOWNLOADER_MIDDLEWARES = {  
    ...  
    'scrapy_zyte_smartproxy.ZyteSmartProxyMiddleware': 610  
}
```

2. Enable the middleware and configure your API key, either through Scrapy settings:

Listing 2: settings.py

```
ZYTE_SMARTPROXY_ENABLED = True  
ZYTE_SMARTPROXY_APIKEY = 'apikey'
```

Or through spider attributes:

```
class MySpider(scrapy.Spider):  
    zyte_smartproxy_enabled = True  
    zyte_smartproxy_apikey = 'apikey'
```

1. Set the ZYTE_SMARTPROXY_URL Scrapy setting as needed:
 - To use the proxy mode of Zyte API, set it to `http://api.zyte.com:8011`:

Listing 3: settings.py

```
ZYTE_SMARTPROXY_URL = "http://api.zyte.com:8011"
```

 **Tip**

This URL is logged, so that you can tell which value was used from crawl logs.

- To use the default Zyte Smart Proxy Manager endpoint, leave it unset.
- To use a custom Zyte Smart Proxy Manager endpoint, in case you have a dedicated or private instance, set it to your custom endpoint. For example:

Listing 4: settings.py

```
ZYTE_SMARTPROXY_URL = "http://myinstance.zyte.com:8011"
```

USAGE

Once the downloader middleware is properly configured, every request goes through the configured Zyte proxy service.

Although the plugin configuration only allows defining a single proxy endpoint and API key, it is possible to override them for specific requests, so that you can use different combinations for different requests within the same spider.

To **override** which combination of endpoint and API key is used for a given request, set `proxy` in the request metadata to a URL indicating both the target endpoint and the API key to use. For example:

```
scrapy.Request(  
    "https://topscrape.com",  
    meta={  
        "proxy": "http://YOUR_API_KEY@api.zyte.com:8011",  
        ...  
    },  
)
```

To **disable** proxying altogether for a given request, set `dont_proxy` to `True` on the request metadata:

```
scrapy.Request(  
    "https://topscrape.com",  
    meta={  
        "dont_proxy": True,  
        ...  
    },  
)
```

You can set [Zyte API proxy headers](#) or [Zyte Smart Proxy Manager headers](#) as regular Scrapy headers, e.g. using the `headers` parameter of `Request` or using the `DEFAULT_REQUEST_HEADERS` setting. For example:

```
scrapy.Request(  
    "https://topscrape.com",  
    headers={  
        "Zyte-Geolocation": "FR",  
        ...  
    },  
)
```

For information about proxy-specific header processing, see [Headers](#).

See also [Settings](#) for the complete list of settings that this downloader middleware supports.