Spider Middleware

The spider middleware is a framework of hooks into Scrapy's spider processing mechanism where you can plug custom functionality to process the responses that are sent to :ref:`topics-spiders` for processing and to process the requests and items that are generated from spiders.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 7`); *backlink*
>
> Unknown interpreted text role "ref".

Activating a spider middleware

To activate a spider middleware component, add it to the :setting:`SPIDER_MIDDLEWARES` setting, which is a dict whose keys are the middleware class path and their values are the middleware orders.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 17`); *backlink*
>
> Unknown interpreted text role "setting".

Here's an example:

```
SPIDER_MIDDLEWARES = {
    'myproject.middlewares.CustomSpiderMiddleware': 543,
}
```

The :setting:`SPIDER_MIDDLEWARES` setting is merged with the :setting:`SPIDER_MIDDLEWARES_BASE` setting defined in Scrapy (and not meant to be overridden) and then sorted by order to get the final sorted list of enabled middlewares: the first middleware is the one closer to the engine and the last is the one closer to the spider. In other words, the :meth:`~scrapy.spidermiddlewares.SpiderMiddleware.process_spider_input` method of each middleware will be invoked in increasing middleware order (100, 200, 300, ...), and the :meth:`~scrapy.spidermiddlewares.SpiderMiddleware.process_spider_output` method of each middleware will be invoked in decreasing order.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 27`); *backlink*
>
> Unknown interpreted text role "setting".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 27`); *backlink*
>
> Unknown interpreted text role "setting".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 27`); *backlink*
>
> Unknown interpreted text role "meth".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 27`); *backlink*
>
> Unknown interpreted text role "meth".

To decide which order to assign to your middleware see the :setting:`SPIDER_MIDDLEWARES_BASE` setting and pick a value according to where you want to insert the middleware. The order does matter because each middleware performs a different action and your middleware could depend on some previous (or subsequent) middleware being applied.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 38`); *backlink*
>
> Unknown interpreted text role "setting".

If you want to disable a builtin middleware (the ones defined in :setting:`SPIDER_MIDDLEWARES_BASE`, and enabled by default) you must define it in your project :setting:`SPIDER_MIDDLEWARES` setting and assign `None` as its value. For example, if you want to disable the off-site middleware:

```
SPIDER_MIDDLEWARES = {
    'myproject.middlewares.CustomSpiderMiddleware': 543,
    'scrapy.spidermiddlewares.offsite.OffsiteMiddleware': None,
}
```

Finally, keep in mind that some middlewares may need to be enabled through a particular setting. See each middleware documentation for more info.

Writing your own spider middleware

Each spider middleware is a Python class that defines one or more of the methods defined below.

The main entry point is the `from_crawler` class method, which receives a :class:`~scrapy.crawler.Crawler` instance. The :class:`~scrapy.crawler.Crawler` object gives you access, for example, to the :ref:`settings <topics-settings>`.

```
.. module:: scrapy.spidermiddlewares
```

```
.. method:: process_spider_input(response, spider)

   This method is called for each response that goes through the spider
   middleware and into the spider, for processing.

   :meth:`process_spider_input` should return ``None`` or raise an
   exception.

   If it returns ``None``, Scrapy will continue processing this response,
   executing all other middlewares until, finally, the response is handed
   to the spider for processing.

   If it raises an exception, Scrapy won't bother calling any other spider
   middleware :meth:`process_spider_input` and will call the request
```

```
errback if there is one, otherwise it will start the :meth:`process_spider_exception`
chain. The output of the errback is chained back in the other
direction for :meth:`process_spider_output` to process it, or
:meth:`process_spider_exception` if it raised an exception.

:param response: the response being processed
:type response: :class:`~scrapy.http.Response` object

:param spider: the spider for which this response is intended
:type spider: :class:`~scrapy.Spider` object
```

```
.. method:: process_spider_output(response, result, spider)

   This method is called with the results returned from the Spider, after
   it has processed the response.

   :meth:`process_spider_output` must return an iterable of
   :class:`~scrapy.Request` objects and :ref:`item object
   <topics-items>`.

   :param response: the response which generated this output from the
     spider
   :type response: :class:`~scrapy.http.Response` object

   :param result: the result returned by the spider
   :type result: an iterable of :class:`~scrapy.Request` objects and
     :ref:`item object <topics-items>`

   :param spider: the spider whose result is being processed
   :type spider: :class:`~scrapy.Spider` object
```

```
.. method:: process_spider_exception(response, exception, spider)

   This method is called when a spider or :meth:`process_spider_output`
   method (from a previous spider middleware) raises an exception.

   :meth:`process_spider_exception` should return either ``None`` or an
   iterable of :class:`~scrapy.Request` or :ref:`item <topics-items>`
   objects.

   If it returns ``None``, Scrapy will continue processing this exception,
   executing any other :meth:`process_spider_exception` in the following
   middleware components, until no middleware components are left and the
   exception reaches the engine (where it's logged and discarded).

   If it returns an iterable the :meth:`process_spider_output` pipeline
   kicks in, starting from the next spider middleware, and no other
   :meth:`process_spider_exception` will be called.

   :param response: the response being processed when the exception was
     raised
   :type response: :class:`~scrapy.http.Response` object

   :param exception: the exception raised
   :type exception: :exc:`Exception` object

   :param spider: the spider which raised the exception
   :type spider: :class:`~scrapy.Spider` object
```

```
.. method:: process_start_requests(start_requests, spider)

   This method is called with the start requests of the spider, and works
   similarly to the :meth:`process_spider_output` method, except that it
   doesn't have a response associated and must return only requests (not
   items).

   It receives an iterable (in the ``start_requests`` parameter) and must
   return another iterable of :class:`~scrapy.Request` objects.

   .. note:: When implementing this method in your spider middleware, you
      should always return an iterable (that follows the input one) and
      not consume all ``start_requests`` iterator because it can be very
      large (or even unbounded) and cause a memory overflow. The Scrapy
      engine is designed to pull start requests while it has capacity to
      process them, so the start requests iterator can be effectively
      endless where there is some other condition for stopping the spider
      (like a time limit or item/page count).

   :param start_requests: the start requests
   :type start_requests: an iterable of :class:`~scrapy.Request`

   :param spider: the spider to whom the start requests belong
   :type spider: :class:`~scrapy.Spider` object
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst`**, line 172)**

Unknown directive type "method".

```
.. method:: from_crawler(cls, crawler)

   If present, this classmethod is called to create a middleware instance
   from a :class:`~scrapy.crawler.Crawler`. It must return a new instance
   of the middleware. Crawler object provides access to all Scrapy core
   components like settings and signals; it is a way for middleware to
   access them and hook its functionality into Scrapy.

   :param crawler: crawler that uses this middleware
   :type crawler: :class:`~scrapy.crawler.Crawler` object
```

Built-in spider middleware reference

This page describes all spider middleware components that come with Scrapy. For information on how to use them and how to write your own spider middleware, see the :ref:`spider middleware usage guide <topics-spider-middleware>`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst`**, line 188);** *backlink*

Unknown interpreted text role "ref".

For a list of the components enabled by default (and their orders) see the :setting:`SPIDER_MIDDLEWARES_BASE` setting.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst`**, line 192);** *backlink*

Unknown interpreted text role "setting".

DepthMiddleware

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst`**, line 198)**

Unknown directive type "module".

```
.. module:: scrapy.spidermiddlewares.depth
   :synopsis: Depth Spider Middleware
```

DepthMiddleware is used for tracking the depth of each Request inside the site being scraped. It works by setting `request.meta['depth'] = 0` whenever there is no value previously set (usually just the first Request) and incrementing it by 1

otherwise.

It can be used to limit the maximum depth to scrape, control Request priority based on their depth, and things like that.

The :class:`DepthMiddleware` can be configured through the following settings (see the settings documentation for more info):

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 211`); *backlink*
>
> Unknown interpreted text role "class".

- :setting:`DEPTH_LIMIT` - The maximum depth that will be allowed to crawl for any site. If zero, no limit will be imposed.

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 214`); *backlink*
  >
  > Unknown interpreted text role "setting".

- :setting:`DEPTH_STATS_VERBOSE` - Whether to collect the number of requests for each depth.

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 216`); *backlink*
  >
  > Unknown interpreted text role "setting".

- :setting:`DEPTH_PRIORITY` - Whether to prioritize the requests based on their depth.

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 218`); *backlink*
  >
  > Unknown interpreted text role "setting".

HttpErrorMiddleware

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 224`)
>
> Unknown directive type "module".
>
> ```
>     .. module:: scrapy.spidermiddlewares.httperror
>        :synopsis: HTTP Error Spider Middleware
> ```

Filter out unsuccessful (erroneous) HTTP responses so that spiders don't have to deal with them, which (most of the time) imposes an overhead, consumes more resources, and makes the spider logic more complex.

According to the HTTP standard, successful responses are those whose status codes are in the 200-300 range.

If you still want to process response codes outside that range, you can specify which response codes the spider is able to handle using the `handle_httpstatus_list` spider attribute or :setting:`HTTPERROR_ALLOWED_CODES` setting.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 238`); *backlink*
>
> Unknown interpreted text role "setting".

For example, if you want your spider to handle 404 responses you can do this:

```
class MySpider(CrawlSpider):
    handle_httpstatus_list = [404]
```

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 249`)
>
> Unknown directive type "reqmeta".

```
   .. reqmeta:: handle_httpstatus_list
```

The `handle_httpstatus_list` key of :attr:`Request.meta <scrapy.Request.meta>` can also be used to specify which response codes to allow on a per-request basis. You can also set the meta key `handle_httpstatus_all` to `True` if you want to allow any response code for a request, and `False` to disable the effects of the `handle_httpstatus_all` key.

Keep in mind, however, that it's usually a bad idea to handle non-200 responses, unless you really know what you're doing.

For more information see: HTTP Status Code Definitions.

HttpErrorMiddleware settings

HTTPERROR_ALLOWED_CODES

Default: `[]`

Pass all responses with non-200 status codes contained in this list.

HTTPERROR_ALLOW_ALL

Default: `False`

Pass all responses, regardless of its status code.

OffsiteMiddleware

Filters out Requests for URLs outside the domains covered by the spider.

This middleware filters out every request whose host names aren't in the spider's :attr:`~scrapy.Spider.allowed_domains` attribute. All subdomains of any domain in the list are also allowed. E.g. the rule `www.example.org` will also allow `bob.www.example.org` but not `www2.example.com` nor `example.com`.

When your spider returns a request for a domain not belonging to those covered by the spider, this middleware will log a debug message similar to this one:

```
DEBUG: Filtered offsite request to 'www.othersite.com': <GET http://www.othersite.com/some/page.html>
```

To avoid filling the log with too much noise, it will only print one of these messages for each new domain filtered. So, for example, if another request for `www.othersite.com` is filtered, no log message will be printed. But if a request for `someothersite.com` is filtered, a message will be printed (but only for the first request filtered).

If the spider doesn't define an :attr:`~scrapy.Spider.allowed_domains` attribute, or the attribute is empty, the offsite middleware will allow all requests.

If the request has the :attr:`~scrapy.Request.dont_filter` attribute set, the offsite middleware will allow the request even if its domain is not listed in allowed domains.

RefererMiddleware

Populates Request `Referer` header, based on the URL of the Response which generated it.

RefererMiddleware settings

REFERER_ENABLED

Default: `True`

Whether to enable referer middleware.

REFERRER_POLICY

Default: `'scrapy.spidermiddlewares.referer.DefaultReferrerPolicy'`

master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 354)

Unknown directive type "reqmeta".

```
.. reqmeta:: referrer_policy
```

Referrer Policy to apply when populating Request "Referer" header.

> **Note**
>
> You can also set the Referrer Policy per request, using the special "`referrer_policy`" :ref:`Request.meta <topics-request-meta>` key, with the same acceptable values as for the REFERRER_POLICY setting.
>
> > **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 359); *backlink***
> >
> > Unknown interpreted text role "ref".

Acceptable values for REFERRER_POLICY

- either a path to a `scrapy.spidermiddlewares.referer.ReferrerPolicy` subclass â€" a custom policy or one of the built-in ones (see classes below),
- or one of the standard W3C-defined string values,
- or the special "`scrapy-default`".

String value	Class name (as a string)
"`scrapy-default`" (default)	:class:`scrapy.spidermiddlewares.referer.DefaultReferrerPolicy` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 375); *backlink*** Unknown interpreted text role "class".
"no-referrer"	:class:`scrapy.spidermiddlewares.referer.NoReferrerPolicy` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 376); *backlink*** Unknown interpreted text role "class".
"no-referrer-when-downgrade"	:class:`scrapy.spidermiddlewares.referer.NoReferrerWhenDowngradePolicy` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 377); *backlink*** Unknown interpreted text role "class".
"same-origin"	:class:`scrapy.spidermiddlewares.referer.SameOriginPolicy` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 378); *backlink*** Unknown interpreted text role "class".

String value	Class name (as a string)
"origin"	:class:`scrapy.spidermiddlewares.referer.OriginPolicy` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 379); backlink`** Unknown interpreted text role "class".
"strict-origin"	:class:`scrapy.spidermiddlewares.referer.StrictOriginPolicy` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 380); backlink`** Unknown interpreted text role "class".
"origin-when-cross-origin"	:class:`scrapy.spidermiddlewares.referer.OriginWhenCrossOriginPolicy` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 381); backlink`** Unknown interpreted text role "class".
"strict-origin-when-cross-origin"	:class:`scrapy.spidermiddlewares.referer.StrictOriginWhenCrossOriginPolicy` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 382); backlink`** Unknown interpreted text role "class".
"unsafe-url"	:class:`scrapy.spidermiddlewares.referer.UnsafeUrlPolicy` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 383); backlink`** Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 385)`

Unknown directive type "autoclass".

```
.. autoclass:: DefaultReferrerPolicy
```

Warning

Scrapy's default referrer policy â€" just like "no-referrer-when-downgrade", the W3C-recommended value for browsers â€" will send a non-empty "Referer" header from any `http(s)://` to any `https://` URL, even if the domain is different.

"same-origin" may be a better choice if you want to remove referrer information for cross-domain requests.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 395)`

Unknown directive type "autoclass".

```
.. autoclass:: NoReferrerPolicy
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 397`)

Unknown directive type "autoclass".

```
.. autoclass:: NoReferrerWhenDowngradePolicy
```

Note

"no-referrer-when-downgrade" policy is the W3C-recommended default, and is used by major web browsers.

However, it is NOT Scrapy's default referrer policy (see :class:`DefaultReferrerPolicy`).

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 402`); *backlink***
>
> Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 404`)

Unknown directive type "autoclass".

```
.. autoclass:: SameOriginPolicy
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 406`)

Unknown directive type "autoclass".

```
.. autoclass:: OriginPolicy
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 408`)

Unknown directive type "autoclass".

```
.. autoclass:: StrictOriginPolicy
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 410`)

Unknown directive type "autoclass".

```
.. autoclass:: OriginWhenCrossOriginPolicy
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 412`)

Unknown directive type "autoclass".

```
.. autoclass:: StrictOriginWhenCrossOriginPolicy
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 414`)

Unknown directive type "autoclass".

```
.. autoclass:: UnsafeUrlPolicy
```

UrlLengthMiddleware

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 432`)

Unknown directive type "module".

```
.. module:: scrapy.spidermiddlewares.urllength
   :synopsis: URL Length Spider Middleware
```

Filters out requests with URLs longer than URLLENGTH_LIMIT

The :class:`UrlLengthMiddleware` can be configured through the following settings (see the settings documentation for more info):

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 439`); *backlink*

Unknown interpreted text role "class".

- :setting:`URLLENGTH_LIMIT` - The maximum URL length to allow for crawled URLs.

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]spider-middleware.rst, line 442`); *backlink*

   Unknown interpreted text role "setting".