Signals

Scrapy uses signals extensively to notify when certain events occur. You can catch some of those signals in your Scrapy project (using an :ref:`extension <topics-extensions>`, for example) to perform additional tasks or extend Scrapy to add functionality not provided out of the box.

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

Even though signals provide several arguments, the handlers that catch them don't need to accept all of them - the signal dispatching mechanism will only deliver the arguments that the handler receives.

You can connect to signals (or send your own) through the :ref:`topics-api-signals`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst`, line 16); *backlink*
>
> Unknown interpreted text role "ref".

Here is a simple example showing how you can catch signals and perform some action:

```python
from scrapy import signals
from scrapy import Spider


class DmozSpider(Spider):
    name = "dmoz"
    allowed_domains = ["dmoz.org"]
    start_urls = [
        "http://www.dmoz.org/Computers/Programming/Languages/Python/Books/",
        "http://www.dmoz.org/Computers/Programming/Languages/Python/Resources/",
    ]


    @classmethod
    def from_crawler(cls, crawler, *args, **kwargs):
        spider = super(DmozSpider, cls).from_crawler(crawler, *args, **kwargs)
        crawler.signals.connect(spider.spider_closed, signal=signals.spider_closed)
        return spider


    def spider_closed(self, spider):
        spider.logger.info('Spider closed: %s', spider.name)


    def parse(self, response):
        pass
```

Deferred signal handlers

Some signals support returning :class:`~twisted.internet.defer.Deferred` objects from their handlers, allowing you to run asynchronous code that does not block Scrapy. If a signal handler returns a :class:`~twisted.internet.defer.Deferred`, Scrapy waits for that :class:`~twisted.internet.defer.Deferred` to fire.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst`, line 53); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst`, line 53); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst`, line 53); *backlink*

Unknown interpreted text role "class".

Let's take an example:

```python
class SignalSpider(scrapy.Spider):
    name = 'signals'
    start_urls = ['https://quotes.toscrape.com/page/1/']

    @classmethod
    def from_crawler(cls, crawler, *args, **kwargs):
        spider = super(SignalSpider, cls).from_crawler(crawler, *args, **kwargs)
        crawler.signals.connect(spider.item_scraped, signal=signals.item_scraped)
        return spider

    def item_scraped(self, item):
        # Send the scraped item to the server
        d = treq.post(
            'http://example.com/post',
            json.dumps(item).encode('ascii'),
            headers={b'Content-Type': [b'application/json']}
        )

        # The next item will be scraped only after
        # deferred (d) is fired
        return d

    def parse(self, response):
        for quote in response.css('div.quote'):
            yield {
                'text': quote.css('span.text::text').get(),
                'author': quote.css('small.author::text').get(),
                'tags': quote.css('div.tags a.tag::text').getall(),
            }
```

See the :ref:`topics-signals-ref` below to know which signals support :class:`~twisted.internet.defer.Deferred`.

> **System Message: ERROR/3 (**D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst**, line 91); *backlink***
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3 (**D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst**, line 91); *backlink***
>
> Unknown interpreted text role "class".

Built-in signals reference

> **System Message: ERROR/3 (**D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst**, line 99)**
>
> Unknown directive type "module".
>
> ```
> .. module:: scrapy.signals
>    :synopsis: Signals definitions
> ```

Here's the list of Scrapy built-in signals and their meaning.

Engine signals

engine_started

> **System Message: ERROR/3 (**D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst**, line 110)**
>
> Unknown directive type "signal".
>
> ```
> .. signal:: engine_started
> ```

> **System Message: ERROR/3 (**D:\onboarding-resources\sample-onboarding-resources\scrapy-**

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-`
`master\docs\topics\[scrapy-master][docs][topics]signals.rst,` **line 111)**

Unknown directive type "function".

```
.. function:: engine_started()

    Sent when the Scrapy engine has started crawling.

    This signal supports returning deferreds from its handlers.
```

Note

This signal may be fired *after* the :signal:`spider_opened` signal, depending on how the spider was started. So **don't** rely on this signal getting fired before :signal:`spider_opened`.

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-`
> `resources\scrapy-master\docs\topics\[scrapy-master][docs]`
> `[topics]signals.rst,` **line 117);** *backlink*
>
> Unknown interpreted text role "signal".

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-`
> `resources\scrapy-master\docs\topics\[scrapy-master][docs]`
> `[topics]signals.rst,` **line 117);** *backlink*
>
> Unknown interpreted text role "signal".

engine_stopped

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-`
`master\docs\topics\[scrapy-master][docs][topics]signals.rst,` **line 124)**

Unknown directive type "signal".

```
.. signal:: engine_stopped
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-`
`master\docs\topics\[scrapy-master][docs][topics]signals.rst,` **line 125)**

Unknown directive type "function".

```
.. function:: engine_stopped()

    Sent when the Scrapy engine is stopped (for example, when a crawling
    process has finished).

    This signal supports returning deferreds from its handlers.
```

Item signals

Note

As at max :setting:`CONCURRENT_ITEMS` items are processed in parallel, many deferreds are fired together using :class:`~twisted.internet.defer.DeferredList`. Hence the next batch waits for the :class:`~twisted.internet.defer.DeferredList` to fire and then runs the respective item signal handler for the next batch of scraped items.

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-`
> `resources\scrapy-master\docs\topics\[scrapy-master][docs]`
> `[topics]signals.rst,` **line 136);** *backlink*
>
> Unknown interpreted text role "setting".

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-`
> `resources\scrapy-master\docs\topics\[scrapy-master][docs]`

item_scraped

```
.. signal:: item_scraped
```

```
.. function:: item_scraped(item, response, spider)

   Sent when an item has been scraped, after it has passed all the
   :ref:`topics-item-pipeline` stages (without being dropped).

   This signal supports returning deferreds from its handlers.

   :param item: the scraped item
   :type item: :ref:`item object <item-types>`

   :param spider: the spider which scraped the item
   :type spider: :class:`~scrapy.Spider` object

   :param response: the response from where the item was scraped
   :type response: :class:`~scrapy.http.Response` object
```

item_dropped

```
.. signal:: item_dropped
```

```
.. function:: item_dropped(item, response, exception, spider)

   Sent after an item has been dropped from the :ref:`topics-item-pipeline`
   when some stage raised a :exc:`~scrapy.exceptions.DropItem` exception.

   This signal supports returning deferreds from its handlers.

   :param item: the item dropped from the :ref:`topics-item-pipeline`
   :type item: :ref:`item object <item-types>`

   :param spider: the spider which scraped the item
   :type spider: :class:`~scrapy.Spider` object

   :param response: the response from where the item was dropped
   :type response: :class:`~scrapy.http.Response` object
```

```
:param exception: the exception (which must be a
    :exc:`~scrapy.exceptions.DropItem` subclass) which caused the item
    to be dropped
:type exception: :exc:`~scrapy.exceptions.DropItem` exception
```

item_error

Spider signals

spider_closed

spider_opened

spider_idle

Note

Scheduling some requests in your :signal:`spider_idle` handler does **not** guarantee that it can prevent the spider from

being closed, although it sometimes can. That's because the spider may still remain idle if all the scheduled requests are rejected by the scheduler (e.g. filtered due to duplication).

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst`, line 283); *backlink*

Unknown interpreted text role "signal".

---

spider_error

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst`, line 292)

Unknown directive type "signal".

```
.. signal:: spider_error
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst`, line 293)

Unknown directive type "function".

```
.. function:: spider_error(failure, response, spider)

    Sent when a spider callback generates an error (i.e. raises an exception).

    This signal does not support returning deferreds from its handlers.

    :param failure: the exception raised
    :type failure: twisted.python.failure.Failure

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

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

Request signals

request_scheduled

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst`, line 314)

Unknown directive type "signal".

```
.. signal:: request_scheduled
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst`, line 315)

Unknown directive type "function".

```
.. function:: request_scheduled(request, spider)

    Sent when the engine schedules a :class:`~scrapy.Request`, to be
    downloaded later.

    This signal does not support returning deferreds from its handlers.

    :param request: the request that reached the scheduler
    :type request: :class:`~scrapy.Request` object

    :param spider: the spider that yielded the request
    :type spider: :class:`~scrapy.Spider` object
```

request_dropped

request_reached_downloader

request_left_downloader

> Sent when a :class:`~scrapy.Request` leaves the downloader, even in case of
> failure.
>
> This signal does not support returning deferreds from its handlers.
>
> :param request: the request that reached the downloader
> :type request: :class:`~scrapy.Request` object
>
> :param spider: the spider that yielded the request
> :type spider: :class:`~scrapy.Spider` object

bytes_received

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst, line 383)

Unknown directive type "versionadded".

```
.. versionadded:: 2.2
```

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

Unknown directive type "signal".

```
.. signal:: bytes_received
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst, line 386)

Unknown directive type "function".

```
.. function:: bytes_received(data, request, spider)

    Sent by the HTTP 1.1 and S3 download handlers when a group of bytes is
    received for a specific request. This signal might be fired multiple
    times for the same request, with partial data each time. For instance,
    a possible scenario for a 25 kb response would be two signals fired
    with 10 kb of data, and a final one with 5 kb of data.

    Handlers for this signal can stop the download of a response while it
    is in progress by raising the :exc:`~scrapy.exceptions.StopDownload`
    exception. Please refer to the :ref:`topics-stop-response-download` topic
    for additional information and examples.

    This signal does not support returning deferreds from its handlers.

    :param data: the data received by the download handler
    :type data: :class:`bytes` object

    :param request: the request that generated the download
    :type request: :class:`~scrapy.Request` object

    :param spider: the spider associated with the response
    :type spider: :class:`~scrapy.Spider` object
```

headers_received

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst, line 413)

Unknown directive type "versionadded".

```
.. versionadded:: 2.5
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst, line 415)

Unknown directive type "signal".

```
.. signal:: headers_received
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst, line 416)

Unknown directive type "function".

```
.. function:: headers_received(headers, body_length, request, spider)

    Sent by the HTTP 1.1 and S3 download handlers when the response headers are
    available for a given request, before downloading any additional content.

    Handlers for this signal can stop the download of a response while it
    is in progress by raising the :exc:`~scrapy.exceptions.StopDownload`
    exception. Please refer to the :ref:`topics-stop-response-download` topic
    for additional information and examples.

    This signal does not support returning deferreds from its handlers.

    :param headers: the headers received by the download handler
    :type headers: :class:`scrapy.http.headers.Headers` object

    :param body_length: expected size of the response body, in bytes
    :type body_length: `int`

    :param request: the request that generated the download
    :type request: :class:`~scrapy.Request` object

    :param spider: the spider associated with the response
    :type spider: :class:`~scrapy.Spider` object
```

Response signals

response_received

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst, line 446)

Unknown directive type "signal".

```
.. signal:: response_received
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst, line 447)

Unknown directive type "function".

```
.. function:: response_received(response, request, spider)

    Sent when the engine receives a new :class:`~scrapy.http.Response` from the
    downloader.

    This signal does not support returning deferreds from its handlers.

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

    :param request: the request that generated the response
    :type request: :class:`~scrapy.Request` object

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

Note

The `request` argument might not contain the original request that reached the downloader, if a :ref:`topics-downloader-middleware` modifies the :class:`~scrapy.http.Response` object and sets a specific `request` attribute.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]signals.rst, line 463); *backlink*

Unknown interpreted text role "ref".

response_downloaded

```
.. signal:: response_downloaded
```

```
.. function:: response_downloaded(response, request, spider)

    Sent by the downloader right after a ``HTTPResponse`` is downloaded.

    This signal does not support returning deferreds from its handlers.

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

    :param request: the request that generated the response
    :type request: :class:`~scrapy.Request` object

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