

Logging

Note

`mod:scrapy.log` has been deprecated alongside its functions in favor of explicit calls to the Python standard logging. Keep reading to learn more about the new logging system.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 8); [backlink](#)
Unknown interpreted text role "mod".

Scrapy uses `mod:logging` for event logging. We'll provide some simple examples to get you started, but for more advanced use-cases it's strongly suggested to read thoroughly its documentation.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 12); [backlink](#)
Unknown interpreted text role "mod".

Logging works out of the box, and can be configured to some extent with the Scrapy settings listed in `ref:topics-logging-settings`.

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

Scrapy calls `func:scrapy.utils.log.configure_logging` to set some reasonable defaults and handle those settings in `ref:topics-logging-settings` when running commands, so it's recommended to manually call it if you're running Scrapy from scripts as described in `ref:run-from-script`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 19); [backlink](#)
Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 19); [backlink](#)
Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 19); [backlink](#)
Unknown interpreted text role "ref".

Log levels

Python's builtin logging defines 5 different levels to indicate the severity of a given log message. Here are the standard ones, listed in decreasing order:

1. `logging.CRITICAL` - for critical errors (highest severity)
2. `logging.ERROR` - for regular errors
3. `logging.WARNING` - for warning messages
4. `logging.INFO` - for informational messages
5. `logging.DEBUG` - for debugging messages (lowest severity)

How to log messages

Here's a quick example of how to log a message using the `logging.WARNING` level:

```
import logging
```

```
logging.warning("This is a warning")
```

There are shortcuts for issuing log messages on any of the standard 5 levels, and there's also a general `logging.log` method which takes a given level as argument. If needed, the last example could be rewritten as:

```
import logging
logging.log(logging.WARNING, "This is a warning")
```

On top of that, you can create different "loggers" to encapsulate messages. (For example, a common practice is to create different loggers for every module). These loggers can be configured independently, and they allow hierarchical constructions.

The previous examples use the root logger behind the scenes, which is a top level logger where all messages are propagated to (unless otherwise specified). Using `logging` helpers is merely a shortcut for getting the root logger explicitly, so this is also an equivalent of the last snippets:

```
import logging
logger = logging.getLogger()
logger.warning("This is a warning")
```

You can use a different logger just by getting its name with the `logging.getLogger` function:

```
import logging
logger = logging.getLogger('mycustomlogger')
logger.warning("This is a warning")
```

Finally, you can ensure having a custom logger for any module you're working on by using the `__name__` variable, which is populated with current module's path:

```
import logging
logger = logging.getLogger(__name__)
logger.warning("This is a warning")
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) logging.rst, line 83)

Unknown directive type "seealso".

```
.. seealso::
```

```
    Module logging, :doc:`HowTo <howto/logging>`
        Basic Logging Tutorial
```

```
    Module logging, :ref:`Loggers <logger>`
        Further documentation on loggers
```

Logging from Spiders

Scrapy provides a `:data:`~scrapy.Spider.logger`` within each Spider instance, which can be accessed and used like this:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) logging.rst, line 96); [backlink](#)

Unknown interpreted text role "data".

```
import scrapy

class MySpider(scrapy.Spider):

    name = 'myspider'
    start_urls = ['https://scrapy.org']

    def parse(self, response):
        self.logger.info('Parse function called on %s', response.url)
```

That logger is created using the Spider's name, but you can use any custom Python logger you want. For example:

```
import logging
import scrapy

logger = logging.getLogger('mycustomlogger')

class MySpider(scrapy.Spider):

    name = 'myspider'
    start_urls = ['https://scrapy.org']
```

```
def parse(self, response):
    logger.info('Parse function called on %s', response.url)
```

Logging configuration

Loggers on their own don't manage how messages sent through them are displayed. For this task, different "handlers" can be attached to any logger instance and they will redirect those messages to appropriate destinations, such as the standard output, files, emails, etc.

By default, Scrapy sets and configures a handler for the root logger, based on the settings below.

Logging settings

These settings can be used to configure the logging:

- `:setting:`LOG_FILE``

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 145); [backlink](#)

Unknown interpreted text role "setting".

- `:setting:`LOG_FILE_APPEND``

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 146); [backlink](#)

Unknown interpreted text role "setting".

- `:setting:`LOG_ENABLED``

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 147); [backlink](#)

Unknown interpreted text role "setting".

- `:setting:`LOG_ENCODING``

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 148); [backlink](#)

Unknown interpreted text role "setting".

- `:setting:`LOG_LEVEL``

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 149); [backlink](#)

Unknown interpreted text role "setting".

- `:setting:`LOG_FORMAT``

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 150); [backlink](#)

Unknown interpreted text role "setting".

- `:setting:`LOG_DATEFORMAT``

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 151); [backlink](#)

Unknown interpreted text role "setting".

- `:setting:'LOG_STDOUT'`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 152); [backlink](#)

Unknown interpreted text role "setting".

- `:setting:'LOG_SHORT_NAMES'`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 153); [backlink](#)

Unknown interpreted text role "setting".

The first couple of settings define a destination for log messages. If `:setting:'LOG_FILE'` is set, messages sent through the root logger will be redirected to a file named `:setting:'LOG_FILE'` with encoding `:setting:'LOG_ENCODING'`. If unset and `:setting:'LOG_ENABLED'` is True, log messages will be displayed on the standard error. If `:setting:'LOG_FILE'` is set and `:setting:'LOG_FILE_APPEND'` is False, the file will be overwritten (discarding the output from previous runs, if any). Lastly, if `:setting:'LOG_ENABLED'` is False, there won't be any visible log output.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 155); [backlink](#)

Unknown interpreted text role "setting".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 155); [backlink](#)

Unknown interpreted text role "setting".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 155); [backlink](#)

Unknown interpreted text role "setting".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 155); [backlink](#)

Unknown interpreted text role "setting".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 155); [backlink](#)

Unknown interpreted text role "setting".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 155); [backlink](#)

Unknown interpreted text role "setting".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 155); [backlink](#)

Unknown interpreted text role "setting".

`:setting:'LOG_LEVEL'` determines the minimum level of severity to display, those messages with lower severity will be filtered out. It ranges through the possible levels listed in [:ref:'topics-logging-levels'](#).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 164); [backlink](#)

Unknown interpreted text role "setting".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 164); [backlink](#)

Unknown interpreted text role "ref".

:setting:`LOG_FORMAT` and :setting:`LOG_DATEFORMAT` specify formatting strings used as layouts for all messages. Those strings can contain any placeholders listed in :ref:`logging's logrecord attributes docs <logrecord-attributes>` and :ref:`datetime's strftime and strptime directives <strftime-strptime-behavior>` respectively.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 168); [backlink](#)

Unknown interpreted text role "setting".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 168); [backlink](#)

Unknown interpreted text role "setting".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 168); [backlink](#)

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 168); [backlink](#)

Unknown interpreted text role "ref".

If :setting:`LOG_SHORT_NAMES` is set, then the logs will not display the Scrapy component that prints the log. It is unset by default, hence logs contain the Scrapy component responsible for that log output.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 174); [backlink](#)

Unknown interpreted text role "setting".

Command-line options

There are command-line arguments, available for all commands, that you can use to override some of the Scrapy settings regarding logging.

- `--logfile FILE`

Overrides :setting:`LOG_FILE`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 185); [backlink](#)

Unknown interpreted text role "setting".

- `--loglevel/-L LEVEL`

Overrides :setting:`LOG_LEVEL`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 187); [backlink](#)

Unknown interpreted text role "setting".

- `--nolog`

Sets :setting:`LOG_ENABLED` to False

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 189); [backlink](#)

Unknown interpreted text role "setting".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 191)

Unknown directive type "seealso".

```
.. seealso::
```

```
Module :mod:`logging.handlers`  
Further documentation on available handlers
```

Custom Log Formats

A custom log format can be set for different actions by extending `:class:`~scrapy.logformatter.LogFormatter`` class and making `:setting:`LOG_FORMATTER`` point to your new class.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 201); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 201); [backlink](#)

Unknown interpreted text role "setting".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 205)

Unknown directive type "autoclass".

```
.. autoclass:: scrapy.logformatter.LogFormatter  
:members:
```

Advanced customization

Because Scrapy uses `stdlib` logging module, you can customize logging using all features of `stdlib` logging.

For example, let's say you're scraping a website which returns many HTTP 404 and 500 responses, and you want to hide all messages like this:

```
2016-12-16 22:00:06 [scrapy.spidermiddlewares.httperror] INFO: Ignoring  
response <500 https://quotes.toscrape.com/page/1-34/>: HTTP status code  
is not handled or not allowed
```

The first thing to note is a logger name - it is in brackets: `[scrapy.spidermiddlewares.httperror]`. If you get just `[scrapy]` then `:setting:`LOG_SHORT_NAMES`` is likely set to `True`; set it to `False` and re-run the crawl.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 224); [backlink](#)

Unknown interpreted text role "setting".

Next, we can see that the message has INFO level. To hide it we should set logging level for `scrapy.spidermiddlewares.httperror` higher than INFO; next level after INFO is WARNING. It could be done e.g. in the spider's `__init__` method:

```
import logging  
import scrapy  
  
class MySpider(scrapy.Spider):  
    # ...  
    def __init__(self, *args, **kwargs):
```

```

logger = logging.getLogger('scrapy.spidermiddlewares.httperror')
logger.setLevel(logging.WARNING)
super().__init__(*args, **kwargs)

```

If you run this spider again then INFO messages from `scrapy.spidermiddlewares.httperror` logger will be gone.

You can also filter log records by `:class:~logging.LogRecord` data. For example, you can filter log records by message content using a substring or a regular expression. Create a `:class:logging.Filter` subclass and equip it with a regular expression pattern to filter out unwanted messages:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 248); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 248); [backlink](#)

Unknown interpreted text role "class".

```

import logging
import re

class ContentFilter(logging.Filter):
    def filter(self, record):
        match = re.search(r'\d{3} [Ee]rror, retrying', record.message)
        if match:
            return False

```

A project-level filter may be attached to the root handler created by Scrapy, this is a wieldy way to filter all loggers in different parts of the project (middlewares, spider, etc.):

```

import logging
import scrapy

class MySpider(scrapy.Spider):
    # ...
    def __init__(self, *args, **kwargs):
        for handler in logging.root.handlers:
            handler.addFilter(ContentFilter())

```

Alternatively, you may choose a specific logger and hide it without affecting other loggers:

```

import logging
import scrapy

class MySpider(scrapy.Spider):
    # ...
    def __init__(self, *args, **kwargs):
        logger = logging.getLogger('my_logger')
        logger.addFilter(ContentFilter())

```

scrapy.utils.log module

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

Unknown directive type "module".

```

.. module:: scrapy.utils.log
   :synopsis: Logging utils

```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) logging.rst, line 295)

Unknown directive type "autofunction".

```

.. autofunction:: configure_logging

``configure_logging`` is automatically called when using Scrapy commands
or :class:`~scrapy.crawler.CrawlerProcess`, but needs to be called explicitly
when running custom scripts using :class:`~scrapy.crawler.CrawlerRunner`.
In that case, its usage is not required but it's recommended.

```

Another option when running custom scripts is to manually configure the logging. To do this you can use `:func:`logging.basicConfig`` to set a basic root handler.

Note that `:class:`~scrapy.crawler.CrawlerProcess`` automatically calls `configure_logging``, so it is recommended to only use `:func:`logging.basicConfig`` together with `:class:`~scrapy.crawler.CrawlerRunner``.

This is an example on how to redirect `INFO`` or higher messages to a file::

```
import logging

logging.basicConfig(
    filename='log.txt',
    format='%(levelname)s: %(message)s',
    level=logging.INFO
)
```

Refer to `:ref:`run-from-script`` for more details about using Scrapy this way.