Feed exports

One of the most frequently required features when implementing scrapers is being able to store the scraped data properly and, quite often, that means generating an "export file" with the scraped data (commonly called "export feed") to be consumed by other systems.

Scrapy provides this functionality out of the box with the Feed Exports, which allows you to generate feeds with the scraped items, using multiple serialization formats and storage backends.

Serialization formats

For serializing the scraped data, the feed exports use the :ref:`Item exporters <topics-exporters>`. These formats are supported out of the box:

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 21);** *backlink*
>
> Unknown interpreted text role "ref".

- :ref:`topics-feed-format-json`

  > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 24);** *backlink*
  >
  > Unknown interpreted text role "ref".

- :ref:`topics-feed-format-jsonlines`

  > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 25);** *backlink*
  >
  > Unknown interpreted text role "ref".

- :ref:`topics-feed-format-csv`

  > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 26);** *backlink*
  >
  > Unknown interpreted text role "ref".

- :ref:`topics-feed-format-xml`

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

But you can also extend the supported format through the :setting:`FEED_EXPORTERS` setting.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 29);** *backlink*
>
> Unknown interpreted text role "setting".

JSON

- Value for the `format` key in the :setting:`FEEDS` setting: `json`

  > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 37);** *backlink*
  >
  > Unknown interpreted text role "setting".

- Exporter used: :class:`~scrapy.exporters.JsonItemExporter`

- See :ref:`this warning <json-with-large-data>` if you're using JSON with large feeds.

JSON lines

- Value for the `format` key in the :setting:`FEEDS` setting: `jsonlines`

- Exporter used: :class:`~scrapy.exporters.JsonLinesItemExporter`

CSV

- Value for the `format` key in the :setting:`FEEDS` setting: `csv`

- Exporter used: :class:`~scrapy.exporters.CsvItemExporter`

- To specify columns to export and their order use :setting:`FEED_EXPORT_FIELDS`. Other feed exporters can also use this option, but it is important for CSV because unlike many other export formats CSV uses a fixed header.

XML

- Value for the `format` key in the :setting:`FEEDS` setting: `xml`

Unknown interpreted text role "setting".

- Exporter used: :class:`~scrapy.exporters.XmlItemExporter`

Pickle

- Value for the `format` key in the :setting:`FEEDS` setting: `pickle`

- Exporter used: :class:`~scrapy.exporters.PickleItemExporter`

Marshal

- Value for the `format` key in the :setting:`FEEDS` setting: `marshal`

- Exporter used: :class:`~scrapy.exporters.MarshalItemExporter`

Storages

When using the feed exports you define where to store the feed using one or multiple URIs (through the :setting:`FEEDS` setting). The feed exports supports multiple storage backend types which are defined by the URI scheme.

The storages backends supported out of the box are:

- :ref:`topics-feed-storage-fs`

- :ref:`topics-feed-storage-ftp`

- :ref:`topics-feed-storage-s3` (requires botocore)

- :ref:`topics-feed-storage-gcs` (requires google-cloud-storage)

- :ref:`topics-feed-storage-stdout`

Some storage backends may be unavailable if the required external libraries are not available. For example, the S3 backend is only available if the botocore library is installed.

Storage URI parameters

The storage URI can also contain parameters that get replaced when the feed is being created. These parameters are:

- `%(time)s` - gets replaced by a timestamp when the feed is being created
- `%(name)s` - gets replaced by the spider name

Any other named parameter gets replaced by the spider attribute of the same name. For example, `%(site_id)s` would get replaced by the `spider.site_id` attribute the moment the feed is being created.

Here are some examples to illustrate:

- Store in FTP using one directory per spider:
  - `ftp://user:password@ftp.example.com/scraping/feeds/%(name)s/%(time)s.json`
- Store in S3 using one directory per spider:
  - `s3://mybucket/scraping/feeds/%(name)s/%(time)s.json`

> **Note**
>
> :ref:`Spider arguments <spiderargs>` become spider attributes, hence they can also be used as storage URI parameters.
>

Storage backends

Local filesystem

The feeds are stored in the local filesystem.

- URI scheme: `file`
- Example URI: `file:///tmp/export.csv`
- Required external libraries: none

Note that for the local filesystem storage (only) you can omit the scheme if you specify an absolute path like `/tmp/export.csv`. This

only works on Unix systems though.

FTP

The feeds are stored in a FTP server.

- URI scheme: `ftp`
- Example URI: `ftp://user:pass@ftp.example.com/path/to/export.csv`
- Required external libraries: none

FTP supports two different connection modes: active or passive. Scrapy uses the passive connection mode by default. To use the active connection mode instead, set the :setting:`FEED_STORAGE_FTP_ACTIVE` setting to `True`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 173); *backlink***
>
> Unknown interpreted text role "setting".

This storage backend uses :ref:`delayed file delivery <delayed-file-delivery>`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 178); *backlink***
>
> Unknown interpreted text role "ref".

S3

The feeds are stored on Amazon S3.

- URI scheme: `s3`
- Example URIs:
  - `s3://mybucket/path/to/export.csv`
  - `s3://aws_key:aws_secret@mybucket/path/to/export.csv`
- Required external libraries: botocore >= 1.4.87

The AWS credentials can be passed as user/password in the URI, or they can be passed through the following settings:

- :setting:`AWS_ACCESS_KEY_ID`

  > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 201); *backlink***
  >
  > Unknown interpreted text role "setting".

- :setting:`AWS_SECRET_ACCESS_KEY`

  > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 202); *backlink***
  >
  > Unknown interpreted text role "setting".

- :setting:`AWS_SESSION_TOKEN` (only needed for temporary security credentials)

  > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 203); *backlink***
  >
  > Unknown interpreted text role "setting".

You can also define a custom ACL and custom endpoint for exported feeds using this setting:

- :setting:`FEED_STORAGE_S3_ACL`

  > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 209); *backlink***
  >
  > Unknown interpreted text role "setting".

- :setting:`AWS_ENDPOINT_URL`

This storage backend uses :ref:`delayed file delivery <delayed-file-delivery>`.

Google Cloud Storage (GCS)

The feeds are stored on Google Cloud Storage.

- URI scheme: `gs`
- Example URIs:
  - `gs://mybucket/path/to/export.csv`
- Required external libraries: google-cloud-storage.

For more information about authentication, please refer to Google Cloud documentation.

You can set a *Project ID* and *Access Control List (ACL)* through the following settings:

- :setting:`FEED_STORAGE_GCS_ACL`

- :setting:`GCS_PROJECT_ID`

This storage backend uses :ref:`delayed file delivery <delayed-file-delivery>`.

Standard output

The feeds are written to the standard output of the Scrapy process.

- URI scheme: `stdout`
- Example URI: `stdout:`
- Required external libraries: none

Delayed file delivery

As indicated above, some of the described storage backends use delayed file delivery.

These storage backends do not upload items to the feed URI as those items are scraped. Instead, Scrapy writes items into a temporary local file, and only once all the file contents have been written (i.e. at the end of the crawl) is that file uploaded to the feed URI.

If you want item delivery to start earlier when using one of these storage backends, use
:setting:`FEED_EXPORT_BATCH_ITEM_COUNT` to split the output items in multiple files, with the specified maximum item count per file. That way, as soon as a file reaches the maximum item count, that file is delivered to the feed URI, allowing item delivery to start way before the end of the crawl.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 269); *backlink*
>
> Unknown interpreted text role "setting".

Item filtering

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 281)
>
> Unknown directive type "versionadded".
>
> ```
> .. versionadded:: 2.6.0
> ```

You can filter items that you want to allow for a particular feed by using the `item_classes` option in :ref:`feeds options <feed-options>`. Only items of the specified types will be added to the feed.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 283); *backlink*
>
> Unknown interpreted text role "ref".

The `item_classes` option is implemented by the :class:`~scrapy.extensions.feedexport.ItemFilter` class, which is the default value of the `item_filter` :ref:`feed option <feed-options>`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 287); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 287); *backlink*
>
> Unknown interpreted text role "ref".

You can create your own custom filtering class by implementing :class:`~scrapy.extensions.feedexport.ItemFilter`'s method `accepts` and taking `feed_options` as an argument.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 290); *backlink*
>
> Unknown interpreted text role "class".

For instance:

```
class MyCustomFilter:

    def __init__(self, feed_options):
        self.feed_options = feed_options

    def accepts(self, item):
        if "field1" in item and item["field1"] == "expected_data":
            return True
        return False
```

You can assign your custom filtering class to the `item_filter` :ref:`option of a feed <feed-options>`. See :setting:`FEEDS` for examples.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 306); *backlink*
>
> Unknown interpreted text role "ref".

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

ItemFilter

```
.. autoclass:: scrapy.extensions.feedexport.ItemFilter
   :members:
```

Post-Processing

```
.. versionadded:: 2.6.0
```

Scrapy provides an option to activate plugins to post-process feeds before they are exported to feed storages. In addition to using :ref:`builtin plugins <builtin-plugins>`, you can create your own :ref:`plugins <custom-plugins>`.

These plugins can be activated through the `postprocessing` option of a feed. The option must be passed a list of post-processing plugins in the order you want the feed to be processed. These plugins can be declared either as an import string or with the imported class of the plugin. Parameters to plugins can be passed through the feed options. See :ref:`feed options <feed-options>` for examples.

Built-in Plugins

```
.. autoclass:: scrapy.extensions.postprocessing.GzipPlugin
```

```
.. autoclass:: scrapy.extensions.postprocessing.LZMAPlugin
```

```
.. autoclass:: scrapy.extensions.postprocessing.Bz2Plugin
```

Custom Plugins

Each plugin is a class that must implement the following methods:

```
.. method:: __init__(self, file, feed_options)

    Initialize the plugin.

    :param file: file-like object having at least the `write`, `tell` and `close` methods implement

    :param feed_options: feed-specific :ref:`options <feed-options>`
    :type feed_options: :class:`dict`
```

```
.. method:: write(self, data)

    Process and write `data` (:class:`bytes` or :class:`memoryview`) into the plugin's target file.
    It must return number of bytes written.
```

```
.. method:: close(self)

    Close the target file object.
```

To pass a parameter to your plugin, use :ref:`feed options <feed-options>`. You can then access those parameters from the __init__ method of your plugin.

Settings

These are the settings used for configuring the feed exports:

- :setting:`FEEDS` (mandatory)

- :setting:`FEED_EXPORT_ENCODING`

- :setting:`FEED_STORE_EMPTY`

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-`
> `resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst,`
> `line 380);` *backlink*
>
> Unknown interpreted text role "setting".

- :setting:`FEED_EXPORT_FIELDS`

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-`
> `resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst,`
> `line 381);` *backlink*
>
> Unknown interpreted text role "setting".

- :setting:`FEED_EXPORT_INDENT`

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-`
> `resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst,`
> `line 382);` *backlink*
>
> Unknown interpreted text role "setting".

- :setting:`FEED_STORAGES`

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-`
> `resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst,`
> `line 383);` *backlink*
>
> Unknown interpreted text role "setting".

- :setting:`FEED_STORAGE_FTP_ACTIVE`

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-`
> `resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst,`
> `line 384);` *backlink*
>
> Unknown interpreted text role "setting".

- :setting:`FEED_STORAGE_S3_ACL`

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-`
> `resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst,`
> `line 385);` *backlink*
>
> Unknown interpreted text role "setting".

- :setting:`FEED_EXPORTERS`

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-`
> `resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst,`
> `line 386);` *backlink*
>
> Unknown interpreted text role "setting".

- :setting:`FEED_EXPORT_BATCH_ITEM_COUNT`

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-`
> `resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst,`
> `line 387);` *backlink*
>
> Unknown interpreted text role "setting".

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-resources\scrapy-`
> `master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst,` `line 389)`
>
> Unknown directive type "currentmodule".
>
> ```
> .. currentmodule:: scrapy.extensions.feedexport
> ```

FEEDS

Default: {}

A dictionary in which every key is a feed URI (or a :class:`pathlib.Path` object) and each value is a nested dictionary containing configuration parameters for the specific feed.

This setting is required for enabling the feed export feature.

See :ref:`topics-feed-storage-backends` for supported URI schemes.

For instance:

```
{
    'items.json': {
        'format': 'json',
        'encoding': 'utf8',
        'store_empty': False,
        'item_classes': [MyItemClass1, 'myproject.items.MyItemClass2'],
        'fields': None,
        'indent': 4,
        'item_export_kwargs': {
           'export_empty_fields': True,
        },
    },
    '/home/user/documents/items.xml': {
        'format': 'xml',
        'fields': ['name', 'price'],
        'item_filter': MyCustomFilter1,
        'encoding': 'latin1',
        'indent': 8,
    },
    pathlib.Path('items.csv.gz'): {
        'format': 'csv',
        'fields': ['price', 'name'],
        'item_filter': 'myproject.filters.MyCustomFilter2',
        'postprocessing': [MyPlugin1, 'scrapy.extensions.postprocessing.GzipPlugin'],
        'gzip_compresslevel': 5,
    },
}
```

The following is a list of the accepted keys and the setting that is used as a fallback value if that key is not provided for a specific feed definition:

- `format`: the :ref:`serialization format <topics-feed-format>`.

This setting is mandatory, there is no fallback value.

- `batch_item_count`: falls back to :setting:`FEED_EXPORT_BATCH_ITEM_COUNT`.

- `encoding`: falls back to :setting:`FEED_EXPORT_ENCODING`.

- `fields`: falls back to :setting:`FEED_EXPORT_FIELDS`.

- `item_classes`: list of :ref:`item classes <topics-items>` to export.

If undefined or empty, all items are exported.

- `item_filter`: a :ref:`filter class <item-filter>` to filter items to export.

:class:`~scrapy.extensions.feedexport.ItemFilter` is used be default.

- `indent`: falls back to :setting:`FEED_EXPORT_INDENT`.

- `item_export_kwargs`: :class:`dict` with keyword arguments for the corresponding :ref:`item exporter class <topics-exporters>`.

- `overwrite`: whether to overwrite the file if it already exists (`True`) or append to its content (`False`).

  The default value depends on the :ref:`storage backend <topics-feed-storage-backends>`:

  - :ref:`topics-feed-storage-fs`: `False`

  - :ref:`topics-feed-storage-ftp`: `True`

> **Note**
>
> Some FTP servers may not support appending to files (the `APPE` FTP command).

- :ref:`topics-feed-storage-s3`: `True` (appending is not supported)

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 487);** *backlink*
>
> Unknown interpreted text role "ref".

- :ref:`topics-feed-storage-stdout`: `False` (overwriting is not supported)

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 490);** *backlink*
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 492)**
>
> Unknown directive type "versionadded".
>
> ```
> .. versionadded:: 2.4.0
> ```

- `store_empty`: falls back to :setting:`FEED_STORE_EMPTY`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 494);** *backlink*
>
> Unknown interpreted text role "setting".

- `uri_params`: falls back to :setting:`FEED_URI_PARAMS`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 496);** *backlink*
>
> Unknown interpreted text role "setting".

- `postprocessing`: list of :ref:`plugins <post-processing>` to use for post-processing.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 498);** *backlink*
>
> Unknown interpreted text role "ref".

The plugins will be used in the order of the list passed.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 502)**
>
> Unknown directive type "versionadded".
>
> ```
> .. versionadded:: 2.6.0
> ```

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 504)**
>
> Unknown directive type "setting".
>
> ```
> .. setting:: FEED_EXPORT_ENCODING
> ```

FEED_EXPORT_ENCODING

Default: `None`

The encoding to be used for the feed.

If unset or set to `None` (default) it uses UTF-8 for everything except JSON output, which uses safe numeric encoding (`\uXXXX` sequences) for historic reasons.

Use `utf-8` if you want UTF-8 for JSON too.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, **line 518**)
>
> Unknown directive type "setting".
>
> ```
>    .. setting:: FEED_EXPORT_FIELDS
> ```

FEED_EXPORT_FIELDS

Default: `None`

A list of fields to export, optional. Example: `FEED_EXPORT_FIELDS = ["foo", "bar", "baz"]`.

Use FEED_EXPORT_FIELDS option to define fields to export and their order.

When FEED_EXPORT_FIELDS is empty or None (default), Scrapy uses the fields defined in :ref:`item objects <topics-items>` yielded by your spider.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, **line 530**); *backlink*
>
> Unknown interpreted text role "ref".

If an exporter requires a fixed set of fields (this is the case for :ref:`CSV <topics-feed-format-csv>` export format) and FEED_EXPORT_FIELDS is empty or None, then Scrapy tries to infer field names from the exported data - currently it uses field names from the first item.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, **line 533**); *backlink*
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, **line 538**)
>
> Unknown directive type "setting".
>
> ```
>    .. setting:: FEED_EXPORT_INDENT
> ```

FEED_EXPORT_INDENT

Default: `0`

Amount of spaces used to indent the output on each level. If `FEED_EXPORT_INDENT` is a non-negative integer, then array elements and object members will be pretty-printed with that indent level. An indent level of `0` (the default), or negative, will put each item on a new line. `None` selects the most compact representation.

Currently implemented only by :class:`~scrapy.exporters.JsonItemExporter` and :class:`~scrapy.exporters.XmlItemExporter`, i.e. when you are exporting to `.json` or `.xml`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, **line 550**); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, **line 550**); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, **line 554**)
>
> Unknown directive type "setting".
>
> ```
>    .. setting:: FEED_STORE_EMPTY
> ```

FEED_STORE_EMPTY

Default: `False`

Whether to export empty feeds (i.e. feeds with no items).

FEED_STORAGES

Default: `{}`

A dict containing additional feed storage backends supported by your project. The keys are URI schemes and the values are paths to storage classes.

FEED_STORAGE_FTP_ACTIVE

Default: `False`

Whether to use the active connection mode when exporting feeds to an FTP server (`True`) or use the passive connection mode instead (`False`, default).

For information about FTP connection modes, see What is the difference between active and passive FTP?.

FEED_STORAGE_S3_ACL

Default: `''` (empty string)

A string containing a custom ACL for feeds exported to Amazon S3 by your project.

For a complete list of available values, access the Canned ACL section on Amazon S3 docs.

FEED_STORAGES_BASE

Default:

```
{
    '': 'scrapy.extensions.feedexport.FileFeedStorage',
    'file': 'scrapy.extensions.feedexport.FileFeedStorage',
    'stdout': 'scrapy.extensions.feedexport.StdoutFeedStorage',
    's3': 'scrapy.extensions.feedexport.S3FeedStorage',
    'ftp': 'scrapy.extensions.feedexport.FTPFeedStorage',
}
```

A dict containing the built-in feed storage backends supported by Scrapy. You can disable any of these backends by assigning `None` to their URI scheme in :setting:`FEED_STORAGES`. E.g., to disable the built-in FTP storage backend (without replacement), place this in your `settings.py`:

```
FEED_STORAGES = {
    'ftp': None,
}
```

FEED_EXPORTERS

Default: `{}`

A dict containing additional exporters supported by your project. The keys are serialization formats and the values are paths to :ref:`Item exporter <topics-exporters>` classes.

FEED_EXPORTERS_BASE

Default:

```
{
    'json': 'scrapy.exporters.JsonItemExporter',
    'jsonlines': 'scrapy.exporters.JsonLinesItemExporter',
    'jl': 'scrapy.exporters.JsonLinesItemExporter',
    'csv': 'scrapy.exporters.CsvItemExporter',
    'xml': 'scrapy.exporters.XmlItemExporter',
    'marshal': 'scrapy.exporters.MarshalItemExporter',
    'pickle': 'scrapy.exporters.PickleItemExporter',
}
```

A dict containing the built-in feed exporters supported by Scrapy. You can disable any of these exporters by assigning `None` to their serialization format in :setting:`FEED_EXPORTERS`. E.g., to disable the built-in CSV exporter (without replacement), place this in your `settings.py`:

```
FEED_EXPORTERS = {
    'csv': None,
}
```

FEED_EXPORT_BATCH_ITEM_COUNT

Default: `0`

If assigned an integer number higher than `0`, Scrapy generates multiple output files storing up to the specified number of items in each output file.

When generating multiple output files, you must use at least one of the following placeholders in the feed URI to indicate how the different output file names are generated:

- `%(batch_time)s` - gets replaced by a timestamp when the feed is being created (e.g. `2020-03-28T14-45-08.237134`)

- `%(batch_id)d` - gets replaced by the 1-based sequence number of the batch.

    Use :ref:`printf-style string formatting <python:old-string-formatting>` to alter the number format. For example, to make the batch ID a 5-digit number by introducing leading zeroes as needed, use `%(batch_id)05d` (e.g `3` becomes `00003`, `123` becomes `00123`).

For instance, if your settings include:

```
FEED_EXPORT_BATCH_ITEM_COUNT = 100
```

And your :command:`crawl` command line is:

```
scrapy crawl spidername -o "dirname/%(batch_id)d-filename%(batch_time)s.json"
```

The command line above can generate a directory tree like:

```
->projectname
-->dirname
--->1-filename2020-03-28T14-45-08.237134.json
--->2-filename2020-03-28T14-45-09.148903.json
--->3-filename2020-03-28T14-45-10.046092.json
```

Where the first and second files contain exactly 100 items. The last one contains 100 items or fewer.

FEED_URI_PARAMS

Default: `None`

A string with the import path of a function to set the parameters to apply with :ref:`printf-style string formatting <python:old-string-formatting>` to the feed URI.

The function signature should be as follows:

Unknown directive type "function".

```
.. function:: uri_params(params, spider)

   Return a :class:`dict` of key-value pairs to apply to the feed URI using
   :ref:`printf-style string formatting <python:old-string-formatting>`.

   :param params: default key-value pairs

       Specifically:

       -   ``batch_id``: ID of the file batch. See
           :setting:`FEED_EXPORT_BATCH_ITEM_COUNT`.

           If :setting:`FEED_EXPORT_BATCH_ITEM_COUNT` is ``0``, ``batch_id``
           is always ``1``.

           .. versionadded:: 2.3.0

       -   ``batch_time``: UTC date and time, in ISO format with ``:``
           replaced with ``-``.

           See :setting:`FEED_EXPORT_BATCH_ITEM_COUNT`.

           .. versionadded:: 2.3.0

       -   ``time``: ``batch_time``, with microseconds set to ``0``.
   :type params: dict

   :param spider: source spider of the feed items
   :type spider: scrapy.Spider

   .. caution:: The function should return a new dictionary, modifying
                the received ``params`` in-place is deprecated.
```

For example, to include the :attr:`name <scrapy.Spider.name>` of the source spider in the feed URI:

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 750); *backlink***
>
> Unknown interpreted text role "attr".

1. Define the following function somewhere in your project:

    ```
    # myproject/utils.py
    def uri_params(params, spider):
        return {**params, 'spider_name': spider.name}
    ```

2. Point :setting:`FEED_URI_PARAMS` to that function in your settings:

    > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)feed-exports.rst`, line 759); *backlink***
    >
    > Unknown interpreted text role "setting".

    ```
    # myproject/settings.py
    FEED_URI_PARAMS = 'myproject.utils.uri_params'
    ```

3. Use `%(spider_name)s` in your feed URI:

    ```
    scrapy crawl <spider_name> -o "%(spider_name)s.jl"
    ```