Downloading and processing files and images

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 7)

Unknown directive type "currentmodule".

```
.. currentmodule:: scrapy.pipelines.images
```

Scrapy provides reusable :doc:`item pipelines </topics/item-pipeline>` for downloading files attached to a particular item (for example, when you scrape products and also want to download their images locally). These pipelines share a bit of functionality and structure (we refer to them as media pipelines), but typically you'll either use the Files Pipeline or the Images Pipeline.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 9); *backlink*

Unknown interpreted text role "doc".

Both pipelines implement these features:

- Avoid re-downloading media that was downloaded recently
- Specifying where to store the media (filesystem directory, FTP server, Amazon S3 bucket, Google Cloud Storage bucket)

The Images Pipeline has a few extra functions for processing images:

- Convert all downloaded images to a common format (JPG) and mode (RGB)
- Thumbnail generation
- Check images width/height to make sure they meet a minimum constraint

The pipelines also keep an internal queue of those media URLs which are currently being scheduled for download, and connect those responses that arrive containing the same media to that queue. This avoids downloading the same media more than once when it's shared by several items.

Using the Files Pipeline

The typical workflow, when using the :class:`FilesPipeline` goes like this:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 35); *backlink*

Unknown interpreted text role "class".

1. In a Spider, you scrape an item and put the URLs of the desired into a `file_urls` field.
2. The item is returned from the spider and goes to the item pipeline.
3. When the item reaches the :class:`FilesPipeline`, the URLs in the `file_urls` field are scheduled for download using the standard Scrapy scheduler and downloader (which means the scheduler and downloader middlewares are reused), but with a higher priority, processing them before other pages are scraped. The item remains "locked" at that particular pipeline stage until the files have finish downloading (or fail for some reason).

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 43); *backlink*

   Unknown interpreted text role "class".

4. When the files are downloaded, another field (`files`) will be populated with the results. This field will contain a list of dicts with information about the downloaded files, such as the downloaded path, the original scraped url (taken from the `file_urls` field), the file checksum and the file status. The files in the list of the `files` field will retain the same order of the original `file_urls` field. If some file failed downloading, an error will be logged and the file won't be present in the `files` field.

Using the Images Pipeline

Using the :class:`ImagesPipeline` is a lot like using the :class:`FilesPipeline`, except the default field names used are different: you use `image_urls` for the image URLs of an item and it will populate an `images` field for the information about the downloaded images.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 64); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 64); *backlink*

Unknown interpreted text role "class".

The advantage of using the :class:`ImagesPipeline` for image files is that you can configure some extra functions like generating thumbnails and filtering the images based on their size.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 69); *backlink*

Unknown interpreted text role "class".

The Images Pipeline requires Pillow 4.0.0 or greater. It is used for thumbnailing and normalizing images to JPEG/RGB format.

Enabling your Media Pipeline

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 84)

Unknown directive type "setting".

```
.. setting:: IMAGES_STORE
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 85)

Unknown directive type "setting".

```
.. setting:: FILES_STORE
```

To enable your media pipeline you must first add it to your project :setting:`ITEM_PIPELINES` setting.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 87); *backlink*

Unknown interpreted text role "setting".

For Images Pipeline, use:

```
ITEM_PIPELINES = {'scrapy.pipelines.images.ImagesPipeline': 1}
```

For Files Pipeline, use:

```
ITEM_PIPELINES = {'scrapy.pipelines.files.FilesPipeline': 1}
```

Note

You can also use both the Files and Images Pipeline at the same time.

Then, configure the target storage setting to a valid value that will be used for storing the downloaded images. Otherwise the pipeline will remain disabled, even if you include it in the :setting:`ITEM_PIPELINES` setting.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 102); *backlink*

Unknown interpreted text role "setting".

For the Files Pipeline, set the :setting:`FILES_STORE` setting:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 106); *backlink*

Unknown interpreted text role "setting".

```
FILES_STORE = '/path/to/valid/dir'
```

For the Images Pipeline, set the :setting:`IMAGES_STORE` setting:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 110); *backlink*

Unknown interpreted text role "setting".

```
IMAGES_STORE = '/path/to/valid/dir'
```

File Naming

Default File Naming

By default, files are stored using an SHA-1 hash of their URLs for the file names.

For example, the following image URL:

```
http://www.example.com/image.jpg
```

Whose `SHA-1 hash` is:

```
3afec3b4765f8f0a07b78f98c07b83f013567a0a
```

Will be downloaded and stored using your chosen :ref:`storage method <topics-supported-storage>` and the following file name:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 132); *backlink*

Unknown interpreted text role "ref".

```
3afec3b4765f8f0a07b78f98c07b83f013567a0a.jpg
```

Custom File Naming

You may wish to use a different calculated file name for saved files. For example, classifying an image by including meta in the file name.

Customize file names by overriding the `file_path` method of your media pipeline.

For example, an image pipeline with image URL:

```
http://www.example.com/product/images/large/front/0000000004166
```

Can be processed into a file name with a condensed hash and the perspective `front`:

```
00b08510e4_front.jpg
```

By overriding `file_path` like this:

```python
import hashlib
from os.path import splitext

def file_path(self, request, response=None, info=None, *, item=None):
    image_url_hash = hashlib.shake_256(request.url.encode()).hexdigest(5)
    image_perspective = request.url.split('/')[-2]
    image_filename = f'{image_url_hash}_{image_perspective}.jpg'

    return image_filename
```

Warning

If your custom file name scheme relies on meta data that can vary between scrapes it may lead to unexpected re-downloading of existing media using new file names.

For example, if your custom file name scheme uses a product title and the site changes an item's product title between scrapes, Scrapy will re-download the same media using updated file names.

For more information about the `file_path` method, see :ref:`topics-media-pipeline-override`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 177); *backlink*
>
> Unknown interpreted text role "ref".

Supported Storage

File system storage

File system storage will save files to the following path:

```
<IMAGES_STORE>/full/<FILE_NAME>
```

Where:

- `<IMAGES_STORE>` is the directory defined in :setting:`IMAGES_STORE` setting for the Images Pipeline.

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 193); *backlink*
  >
  > Unknown interpreted text role "setting".

- `full` is a sub-directory to separate full images from thumbnails (if used). For more info see :ref:`topics-images-thumbnails`.

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 196); *backlink*
  >
  > Unknown interpreted text role "ref".

- `<FILE_NAME>` is the file name assigned to the file. For more info see :ref:`topics-file-naming`.

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 199); *backlink*
  >
  > Unknown interpreted text role "ref".

FTP server storage

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 207)
>
> Unknown directive type "versionadded".
>
> ```
> .. versionadded:: 2.0
> ```

:setting:`FILES_STORE` and :setting:`IMAGES_STORE` can point to an FTP server. Scrapy will automatically upload the files to the server.

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

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

:setting:`FILES_STORE` and :setting:`IMAGES_STORE` should be written in one of the following forms:

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

master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 212); *backlink*

Unknown interpreted text role "setting".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 212); *backlink*

Unknown interpreted text role "setting".

```
ftp://username:password@address:port/path
ftp://address:port/path
```

If `username` and `password` are not provided, they are taken from the :setting:`FTP_USER` and :setting:`FTP_PASSWORD` settings respectively.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 218); *backlink*

Unknown interpreted text role "setting".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 218); *backlink*

Unknown interpreted text role "setting".

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)media-pipeline.rst, line 221); *backlink*

Unknown interpreted text role "setting".

Amazon S3 storage

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 230)

Unknown directive type "setting".

```
.. setting:: FILES_STORE_S3_ACL
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 231)

Unknown directive type "setting".

```
.. setting:: IMAGES_STORE_S3_ACL
```

If botocore >= 1.4.87 is installed, :setting:`FILES_STORE` and :setting:`IMAGES_STORE` can represent an Amazon S3 bucket. Scrapy will automatically upload the files to the bucket.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 233); *backlink*

Unknown interpreted text role "setting".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 233); *backlink*

Unknown interpreted text role "setting".

For example, this is a valid :setting:`IMAGES_STORE` value:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 237); *backlink*

Unknown interpreted text role "setting".

```
IMAGES_STORE = 's3://bucket/images'
```

You can modify the Access Control List (ACL) policy used for the stored files, which is defined by the :setting:`FILES_STORE_S3_ACL` and :setting:`IMAGES_STORE_S3_ACL` settings. By default, the ACL is set to `private`. To make the files publicly available use the `public-read` policy:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 241); *backlink*

Unknown interpreted text role "setting".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 241); *backlink*

Unknown interpreted text role "setting".

```
IMAGES_STORE_S3_ACL = 'public-read'
```

For more information, see canned ACLs in the Amazon S3 Developer Guide.

You can also use other S3-like storages. Storages like self-hosted Minio or s3.scality. All you need to do is set endpoint option in you Scrapy settings:

```
AWS_ENDPOINT_URL = 'http://minio.example.com:9000'
```

For self-hosting you also might feel the need not to use SSL and not to verify SSL connection:

```
AWS_USE_SSL = False # or True (None by default)
AWS_VERIFY = False # or True (None by default)
```

Google Cloud Storage

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 273)

Unknown directive type "setting".

```
.. setting:: FILES_STORE_GCS_ACL
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 274)

Unknown directive type "setting".

```
.. setting:: IMAGES_STORE_GCS_ACL
```

:setting:`FILES_STORE` and :setting:`IMAGES_STORE` can represent a Google Cloud Storage bucket. Scrapy will automatically upload the files to the bucket. (requires google-cloud-storage)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 276); *backlink*

Unknown interpreted text role "setting".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 276); *backlink*

Unknown interpreted text role "setting".

For example, these are valid :setting:`IMAGES_STORE` and :setting:`GCS_PROJECT_ID` settings:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 281); *backlink*

Unknown interpreted text role "setting".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master) (docs) (topics)media-pipeline.rst`, **line 281**); *backlink*

Unknown interpreted text role "setting".

```
IMAGES_STORE = 'gs://bucket/images/'
GCS_PROJECT_ID = 'project_id'
```

For information about authentication, see this documentation.

You can modify the Access Control List (ACL) policy used for the stored files, which is defined by the :setting:`FILES_STORE_GCS_ACL` and :setting:`IMAGES_STORE_GCS_ACL` settings. By default, the ACL is set to `''` (empty string) which means that Cloud Storage applies the bucket's default object ACL to the object. To make the files publicly available use the `publicRead` policy:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master) (docs) (topics)media-pipeline.rst`, **line 290**); *backlink*

Unknown interpreted text role "setting".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master) (docs) (topics)media-pipeline.rst`, **line 290**); *backlink*

Unknown interpreted text role "setting".

```
IMAGES_STORE_GCS_ACL = 'publicRead'
```

For more information, see Predefined ACLs in the Google Cloud Platform Developer Guide.

Usage example

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master) (docs) (topics)media-pipeline.rst`, **line 306**)

Unknown directive type "setting".

```
.. setting:: FILES_URLS_FIELD
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master) (docs) (topics)media-pipeline.rst`, **line 307**)

Unknown directive type "setting".

```
.. setting:: FILES_RESULT_FIELD
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master) (docs) (topics)media-pipeline.rst`, **line 308**)

Unknown directive type "setting".

```
.. setting:: IMAGES_URLS_FIELD
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master) (docs) (topics)media-pipeline.rst`, **line 309**)

Unknown directive type "setting".

```
.. setting:: IMAGES_RESULT_FIELD
```

In order to use a media pipeline, first :ref:`enable it <topics-media-pipeline-enabling>`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master) (docs) (topics)media-pipeline.rst`, **line 311**); *backlink*

Unknown interpreted text role "ref".

Then, if a spider returns an :ref:`item object <topics-items>` with the URLs field (`file_urls` or `image_urls`, for the Files or Images Pipeline respectively), the pipeline will put the results under the respective field (`files` or `images`).

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 314); *backlink*

Unknown interpreted text role "ref".

When using :ref:`item types <item-types>` for which fields are defined beforehand, you must define both the URLs field and the results field. For example, when using the images pipeline, items must define both the `image_urls` and the `images` field. For instance, using the :class:`~scrapy.Item` class:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 319); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 319); *backlink*

Unknown interpreted text role "class".

```
import scrapy

class MyItem(scrapy.Item):
    # ... other item fields ...
    image_urls = scrapy.Field()
    images = scrapy.Field()
```

If you want to use another field name for the URLs key or for the results key, it is also possible to override it.

For the Files Pipeline, set :setting:`FILES_URLS_FIELD` and/or :setting:`FILES_RESULT_FIELD` settings:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 334); *backlink*

Unknown interpreted text role "setting".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 334); *backlink*

Unknown interpreted text role "setting".

```
FILES_URLS_FIELD = 'field_name_for_your_files_urls'
FILES_RESULT_FIELD = 'field_name_for_your_processed_files'
```

For the Images Pipeline, set :setting:`IMAGES_URLS_FIELD` and/or :setting:`IMAGES_RESULT_FIELD` settings:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 340); *backlink*

Unknown interpreted text role "setting".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 340); *backlink*

Unknown interpreted text role "setting".

```
IMAGES_URLS_FIELD = 'field_name_for_your_images_urls'
IMAGES_RESULT_FIELD = 'field_name_for_your_processed_images'
```

If you need something more complex and want to override the custom pipeline behaviour, see :ref:`topics-media-pipeline-override`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 346); *backlink*

Unknown interpreted text role "ref".

If you have multiple image pipelines inheriting from ImagePipeline and you want to have different settings in different pipelines you can set setting keys preceded with uppercase name of your pipeline class. E.g. if your pipeline is called MyPipeline and you want to have custom IMAGES_URLS_FIELD you define setting MYPIPELINE_IMAGES_URLS_FIELD and your custom settings will be used.

Additional features

File expiration

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 364)

Unknown directive type "setting".

```
.. setting:: IMAGES_EXPIRES
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 365)

Unknown directive type "setting".

```
.. setting:: FILES_EXPIRES
```

The Image Pipeline avoids downloading files that were downloaded recently. To adjust this retention delay use the :setting:`FILES_EXPIRES` setting (or :setting:`IMAGES_EXPIRES`, in case of Images Pipeline), which specifies the delay in number of days:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 367); *backlink*

Unknown interpreted text role "setting".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 367); *backlink*

Unknown interpreted text role "setting".

```
# 120 days of delay for files expiration
FILES_EXPIRES = 120

# 30 days of delay for images expiration
IMAGES_EXPIRES = 30
```

The default value for both settings is 90 days.

If you have pipeline that subclasses FilesPipeline and you'd like to have different setting for it you can set setting keys preceded by uppercase class name. E.g. given pipeline class called MyPipeline you can set setting key:

MYPIPELINE_FILES_EXPIRES = 180

and pipeline class MyPipeline will have expiration time set to 180.

The last modified time from the file is used to determine the age of the file in days, which is then compared to the set expiration time to determine if the file is expired.

Thumbnail generation for images

The Images Pipeline can automatically create thumbnails of the downloaded images.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 399)

Unknown directive type "setting".

```
.. setting:: IMAGES_THUMBS
```

In order to use this feature, you must set :setting:`IMAGES_THUMBS` to a dictionary where the keys are the thumbnail names and the values are their dimensions.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 401); *backlink*

Unknown interpreted text role "setting".

For example:

```
IMAGES_THUMBS = {
    'small': (50, 50),
    'big': (270, 270),
}
```

When you use this feature, the Images Pipeline will create thumbnails of the each specified size with this format:

```
<IMAGES_STORE>/thumbs/<size_name>/<image_id>.jpg
```

Where:

- `<size_name>` is the one specified in the :setting:`IMAGES_THUMBS` dictionary keys (`small`, `big`, etc)

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 418); *backlink*

  Unknown interpreted text role "setting".

- `<image_id>` is the SHA-1 hash of the image url

Example of image files stored using `small` and `big` thumbnail names:

```
<IMAGES_STORE>/full/63bbfea82b8880ed33cdb762aa11fab722a90a24.jpg
<IMAGES_STORE>/thumbs/small/63bbfea82b8880ed33cdb762aa11fab722a90a24.jpg
<IMAGES_STORE>/thumbs/big/63bbfea82b8880ed33cdb762aa11fab722a90a24.jpg
```

The first one is the full image, as downloaded from the site.

Filtering out small images

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 436)

Unknown directive type "setting".

```
.. setting:: IMAGES_MIN_HEIGHT
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 438)

Unknown directive type "setting".

```
.. setting:: IMAGES_MIN_WIDTH
```

When using the Images Pipeline, you can drop images which are too small, by specifying the minimum allowed size in the :setting:`IMAGES_MIN_HEIGHT` and :setting:`IMAGES_MIN_WIDTH` settings.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 440); *backlink*

Unknown interpreted text role "setting".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 440); *backlink*

Unknown interpreted text role "setting".

For example:

```
IMAGES_MIN_HEIGHT = 110
IMAGES_MIN_WIDTH = 110
```

Note

The size constraints don't affect thumbnail generation at all.

It is possible to set just one size constraint or both. When setting both of them, only images that satisfy both minimum sizes will be saved. For the above example, images of sizes (105 x 105) or (105 x 200) or (200 x 105) will all be dropped because at least one dimension is shorter than the constraint.

By default, there are no size constraints, so all images are processed.

Allowing redirections

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 462)

Unknown directive type "setting".

```
.. setting:: MEDIA_ALLOW_REDIRECTS
```

By default media pipelines ignore redirects, i.e. an HTTP redirection to a media file URL request will mean the media download is considered failed.

To handle media redirections, set this setting to `True`:

```
MEDIA_ALLOW_REDIRECTS = True
```

Extending the Media Pipelines

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 476)

Unknown directive type "module".

```
.. module:: scrapy.pipelines.files
   :synopsis: Files Pipeline
```

See here the methods that you can override in your custom Files Pipeline:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 483)

Unknown directive type "method".

```
.. method:: file_path(self, request, response=None, info=None, *, item=None)

   This method is called once per downloaded item. It returns the
   download path of the file originating from the specified
   :class:`response <scrapy.http.Response>`.

   In addition to ``response``, this method receives the original
   :class:`request <scrapy.Request>`,
   :class:`info <scrapy.pipelines.media.MediaPipeline.SpiderInfo>` and
   :class:`item <scrapy.Item>`

   You can override this method to customize the download path of each file.

   For example, if file URLs end like regular paths (e.g.
   ``https://example.com/a/b/c/foo.png``), you can use the following
   approach to download all files into the ``files`` folder with their
   original filenames (e.g. ``files/foo.png``)::

     import os
     from urllib.parse import urlparse

     from scrapy.pipelines.files import FilesPipeline

     class MyFilesPipeline(FilesPipeline):

         def file_path(self, request, response=None, info=None, *, item=None):
             return 'files/' + os.path.basename(urlparse(request.url).path)

   Similarly, you can use the ``item`` to determine the file path based on some item
   property.
   By default the :meth:`file_path` method returns
   ``full/<request URL hash>.<extension>``.

   .. versionadded:: 2.4
      The *item* parameter.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 520)

Unknown directive type "method".

```
.. method:: FilesPipeline.get_media_requests(item, info)

   As seen on the workflow, the pipeline will get the URLs of the images to
   download from the item. In order to do this, you can override the
   :meth:`~get_media_requests` method and return a Request for each
   file URL::

      from itemadapter import ItemAdapter

      def get_media_requests(self, item, info):
          adapter = ItemAdapter(item)
          for file_url in adapter['file_urls']:
              yield scrapy.Request(file_url)

   Those requests will be processed by the pipeline and, when they have finished
   downloading, the results will be sent to the
   :meth:`~item_completed` method, as a list of 2-element tuples.
   Each tuple will contain ``(success, file_info_or_error)`` where:

   * ``success`` is a boolean which is ``True`` if the image was downloaded
     successfully or ``False`` if it failed for some reason

   * ``file_info_or_error`` is a dict containing the following keys (if
     success is ``True``) or a :exc:`~twisted.python.failure.Failure` if
     there was a problem.

     * ``url`` - the url where the file was downloaded from. This is the url of
       the request returned from the :meth:`~get_media_requests`
       method.

     * ``path`` - the path (relative to :setting:`FILES_STORE`) where the file
       was stored

     * ``checksum`` - a `MD5 hash`_ of the image contents

     * ``status`` - the file status indication.

       .. versionadded:: 2.2

       It can be one of the following:

       * ``downloaded`` - file was downloaded.
       * ``uptodate`` - file was not downloaded, as it was downloaded recently,
         according to the file expiration policy.
       * ``cached`` - file was already scheduled for download, by another item
         sharing the same file.

   The list of tuples received by :meth:`~item_completed` is
   guaranteed to retain the same order of the requests returned from the
   :meth:`~get_media_requests` method.

   Here's a typical value of the ``results`` argument::

       [(True,
         {'checksum': '2b00042f7481c7b056c4b410d28f33cf',
          'path': 'full/0a79c461a4062ac383dc4fade7bc09f1384a3910.jpg',
          'url': 'http://www.example.com/files/product1.pdf',
          'status': 'downloaded'}),
        (False,
         Failure(...))]

   By default the :meth:`get_media_requests` method returns ``None`` which
   means there are no files to download for the item.
```

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

master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 584)

Unknown directive type "method".

```
.. method:: FilesPipeline.item_completed(results, item, info)

   The :meth:`FilesPipeline.item_completed` method called when all file
   requests for a single item have completed (either finished downloading, or
   failed for some reason).

   The :meth:`~item_completed` method must return the
   output that will be sent to subsequent item pipeline stages, so you must
   return (or drop) the item, as you would in any pipeline.

   Here is an example of the :meth:`~item_completed` method where we
   store the downloaded file paths (passed in results) in the ``file_paths``
   item field, and we drop the item if it doesn't contain any files::

       from itemadapter import ItemAdapter
       from scrapy.exceptions import DropItem

       def item_completed(self, results, item, info):
           file_paths = [x['path'] for ok, x in results if ok]
           if not file_paths:
               raise DropItem("Item contains no files")
           adapter = ItemAdapter(item)
           adapter['file_paths'] = file_paths
           return item

   By default, the :meth:`item_completed` method returns the item.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 612)

Unknown directive type "module".

```
.. module:: scrapy.pipelines.images
   :synopsis: Images Pipeline
```

See here the methods that you can override in your custom Images Pipeline:

The :class:`ImagesPipeline` is an extension of the :class:`FilesPipeline`, customizing the field names and adding custom behavior for images.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 619); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 619); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst, line 622)

Unknown directive type "method".

```
.. method:: file_path(self, request, response=None, info=None, *, item=None)

   This method is called once per downloaded item. It returns the
   download path of the file originating from the specified
   :class:`response <scrapy.http.Response>`.

   In addition to ``response``, this method receives the original
   :class:`request <scrapy.Request>`,
   :class:`info <scrapy.pipelines.media.MediaPipeline.SpiderInfo>` and
   :class:`item <scrapy.Item>`

   You can override this method to customize the download path of each file.
```

```
For example, if file URLs end like regular paths (e.g.
``https://example.com/a/b/c/foo.png``), you can use the following
approach to download all files into the ``files`` folder with their
original filenames (e.g. ``files/foo.png``)::

  import os
  from urllib.parse import urlparse

  from scrapy.pipelines.images import ImagesPipeline

  class MyImagesPipeline(ImagesPipeline):

      def file_path(self, request, response=None, info=None, *, item=None):
          return 'files/' + os.path.basename(urlparse(request.url).path)

Similarly, you can use the ``item`` to determine the file path based on some item
property.

By default the :meth:`file_path` method returns
``full/<request URL hash>.<extension>``.

.. versionadded:: 2.4
   The *item* parameter.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 659)

Unknown directive type "method".

```
.. method:: ImagesPipeline.get_media_requests(item, info)

   Works the same way as :meth:`FilesPipeline.get_media_requests` method,
   but using a different field name for image urls.

   Must return a Request for each image URL.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 666)

Unknown directive type "method".

```
.. method:: ImagesPipeline.item_completed(results, item, info)

   The :meth:`ImagesPipeline.item_completed` method is called when all image
   requests for a single item have completed (either finished downloading, or
   failed for some reason).

   Works the same way as :meth:`FilesPipeline.item_completed` method,
   but using a different field names for storing image downloading results.

   By default, the :meth:`item_completed` method returns the item.
```

Custom Images pipeline example

Here is a full example of the Images Pipeline whose methods are exemplified above:

```
import scrapy
from itemadapter import ItemAdapter
from scrapy.exceptions import DropItem
from scrapy.pipelines.images import ImagesPipeline

class MyImagesPipeline(ImagesPipeline):

    def get_media_requests(self, item, info):
        for image_url in item['image_urls']:
            yield scrapy.Request(image_url)

    def item_completed(self, results, item, info):
        image_paths = [x['path'] for ok, x in results if ok]
        if not image_paths:
            raise DropItem("Item contains no images")
        adapter = ItemAdapter(item)
        adapter['image_paths'] = image_paths
        return item
```

To enable your custom media pipeline component you must add its class import path to the :setting:`ITEM_PIPELINES` setting, like in the following example:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)media-pipeline.rst`, line 706); *backlink*

Unknown interpreted text role "setting".

```
ITEM_PIPELINES = {
    'myproject.pipelines.MyImagesPipeline': 300
}
```