饲料出口

版本0.10中的新功能。

实现刮刀时最常需要的功能之一是能够正确存储刮削数据，并且通常，这意味着生成带有刮削数据（通常称为“导出进给”）的“导出文件”，供其他系统使用。

Scrapy通过Feed Exports提供开箱即用的功能，允许您使用多种序列化格式和存储后端生成带有已删除项目的Feed。

序列化格式

对于序列化已删除的数据，Feed导出使用项目导出器。开箱即用支持这些格式：

- JSON
- JSON行
- CSV
- XML

但您也可以通过`FEED_EXPORTERS`设置扩展支持的格式。

JSON

- `FEED_FORMAT`：`json`
- 出口商使用：`JsonItemExporter`
- 如果您将JSON与大型Feed一起使用，请参阅此警告。

JSON行

- `FEED_FORMAT`：`jsonlines`
- 出口商使用：`JsonLinesItemExporter`

CSV

- `FEED_FORMAT`：`csv`
- 出口商使用：`CsvItemExporter`
- 指定要导出的列及其使用顺序`FEED_EXPORT_FIELDS`。其他Feed导出器也可以使用此选项，但它对CSV很重要，因为与许多其他导出格式不同，CSV使用固定标头。

XML

- `FEED_FORMAT` ： `xml`
- 出口商使用： `XmlItemExporter`

泡菜

- `FEED_FORMAT` ： `pickle`
- 出口商使用： `PickleItemExporter`

元帅

- `FEED_FORMAT` ： `marshal`
- 出口商使用： `MarshalItemExporter`

存储

使用Feed导出时，您可以使用URI（通过`FEED_URI`设置）定义存储Feed的位置。Feed导出支持多种存储后端类型，这些类型由URI方案定义。

支持开箱即用的存储后端是：

- 本地文件系统
- FTP
- S3（需要 botocore或 boto ）
- 标准输出

如果所需的外部库不可用，则某些存储后端可能不可用。例如，S3后端仅在安装了botocore或boto库时才可用（Scrapy 仅在Python 2上支持boto）。

存储URI参数

存储URI还可以包含在创建订阅源时替换的参数。这些参数是：

- `%(time)s` - 在创建订阅源时，将替换为时间戳
- `%(name)s` - 被蜘蛛名称取代

任何其他命名参数都会被同名的spider属性替换。例如， 在创建Feed时将被属性`%(site_id)s`替换`spider.site_id`。

以下是一些示例：

- 使用每个蜘蛛一个目录存储在FTP中：
  - `ftp://user:password@ftp.example.com/scraping/feeds/%(name)s/%(time)s.json`
- 使用每个蜘蛛一个目录存储在S3中：
  - `s3://mybucket/scraping/feeds/%(name)s/%(time)s.json`

存储后端

本地文件系统

订阅源存储在本地文件系统中。

- URI方案： `file`
- 示例URI： `file:///tmp/export.csv`
- 必需的外部库：无

请注意，对于本地文件系统存储（仅限），如果指定类似的绝对路径，则可以省略该方案 `/tmp/export.csv`。这仅适用于Unix系统。

FTP

订阅源存储在FTP服务器中。

- URI方案： `ftp`
- 示例URI： `ftp://user:pass@ftp.example.com/path/to/export.csv`
- 必需的外部库：无

S3

Feed存储在Amazon S3上。

- URI方案： `s3`
- 示例URI：
  - `s3://mybucket/path/to/export.csv`
  - `s3://aws_key:aws_secret@mybucket/path/to/export.csv`
- 必需的外部库：botocore（Python 2和Python 3）或boto（仅限Python 2）

AWS凭证可以作为URI中的用户/密码传递，也可以通过以下设置传递：

- `AWS_ACCESS_KEY_ID`
- `AWS_SECRET_ACCESS_KEY`

标准输出

Feed被写入Scrapy流程的标准输出。

- URI方案： `stdout`
- 示例URI： `stdout:`
- 必需的外部库：无

设置

这些是用于配置Feed导出的设置：

- `FEED_URI` （强制）
- `FEED_FORMAT`
- `FEED_STORAGES`
- `FEED_EXPORTERS`
- `FEED_STORE_EMPTY`
- `FEED_EXPORT_ENCODING`
- `FEED_EXPORT_FIELDS`
- `FEED_EXPORT_INDENT`

FEED_URI

默认： `None`

导出Feed的URI。有关支持的URI方案，请参阅存储后端。

启用Feed导出时需要此设置。

FEED_FORMAT

用于Feed的序列化格式。请参阅 序列化格式以获取可能的值。

FEED_EXPORT_ENCODING

默认： `None`

用于Feed的编码。

如果未设置或设置为 `None` （默认），它将使用UTF-8用于除JSON输出之外的所有内容，JSON输出使用安全数字编码（ `\uXXXX` 序列）以用于历史原因。

`utf-8` 如果您也想要UTF-8用于JSON，请使用。

FEED_EXPORT_FIELDS

默认： `None`

要导出的字段列表，可选。示例：。`FEED_EXPORT_FIELDS = ["foo", "bar", "baz"]`

使用FEED_EXPORT_FIELDS选项定义要导出的字段及其顺序。

当FEED_EXPORT_FIELDS为空或无（默认）时，Scrapy使用`Item`蜘蛛正在产生的dicts或子类中定义的字段。

如果导出器需要一组固定的字段（这是CSV导出格式的情况 ）并且FEED_EXPORT_FIELDS为空或无，则Scrapy会尝试从导出的数据中推断字段名称 - 目前它使用第一个项目中的字段名称。

FEED_EXPORT_INDENT

默认： `0`

用于在每个级别上缩进输出的空间量。如果`FEED_EXPORT_INDENT`是非负整数，则数组元素和对象成员将使用该缩进级别进行漂亮打印。缩进级别`0`（默认值）或负数将把每个项目放在一个新行上。`None`选择最紧凑的表示。

目前仅由`JsonItemExporter`和实施`XmlItemExporter`，即当您导出到`.json`或`.xml`。

FEED_STORE_EMPTY

默认： `False`

是否导出空的Feed（即没有项目的Feed）。

FEED_STORAGES

默认： `{}`

包含项目支持的其他Feed存储后端的dict。键是URI方案，值是存储类的路径。

FEED_STORAGES_BASE

默认：

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

包含Scrapy支持的内置Feed存储后端的dict。您可以通过分配`None`其URI方案 来禁用任何这些后端`FEED_STORAGES`。例如，要禁用内置FTP存储后端（无需替换），请将其放在`settings.py`：

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

FEED_EXPORTERS

默认：`{}`

包含项目支持的其他导出器的dict。键是序列化格式，值是Item导出器类的路径。

FEED_EXPORTERS_BASE

默认：

```
{
    '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',
}
```

包含Scrapy支持的内置Feed导出器的dict。您可以通过分配`None`其序列化格式来禁用任何这些导出器`FEED_EXPORTERS`。例如，要禁用内置CSV导出器（无需替换），请将其放入`settings.py`：

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