

Item Exporters

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

Unknown directive type "module".

```
.. module:: scrapy.exporters
   :synopsis: Item Exporters
```

Once you have scraped your items, you often want to persist or export those items, to use the data in some other application. That is, after all, the whole purpose of the scraping process.

For this purpose Scrapy provides a collection of Item Exporters for different output formats, such as XML, CSV or JSON.

Using Item Exporters

If you are in a hurry, and just want to use an Item Exporter to output scraped data see the [ref`topics-feed-exports`](#). Otherwise, if you want to know how Item Exporters work or need more custom functionality (not covered by the default exports), continue reading below.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 20); [backlink](#)

Unknown interpreted text role "ref".

In order to use an Item Exporter, you must instantiate it with its required args. Each Item Exporter requires different arguments, so check each exporter documentation to be sure, in [ref`topics-exporters-reference`](#). After you have instantiated your exporter, you have to:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 25); [backlink](#)

Unknown interpreted text role "ref".

1. call the method `meth:`~BaseItemExporter.start_exporting`` in order to signal the beginning of the exporting process

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 30); [backlink](#)

Unknown interpreted text role "meth".

2. call the `meth:`~BaseItemExporter.export_item`` method for each item you want to export

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 33); [backlink](#)

Unknown interpreted text role "meth".

3. and finally call the `meth::~BaseItemExporter.finish_exporting`` to signal the end of the exporting process

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 36); [backlink](#)

Unknown interpreted text role "meth".

Here you can see an `doc::Item Pipeline <item-pipeline>` which uses multiple Item Exporters to group scraped items to different files according to the value of one of their fields:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 39); [backlink](#)

Unknown interpreted text role "doc".

```

from itemadapter import ItemAdapter
from scrapy.exporters import XmlItemExporter

class PerYearXmlExportPipeline:
    """Distribute items across multiple XML files according to their 'year' field"""

    def open_spider(self, spider):
        self.year_to_exporter = {}

    def close_spider(self, spider):
        for exporter, xml_file in self.year_to_exporter.values():
            exporter.finish_exporting()
            xml_file.close()

    def _exporter_for_item(self, item):
        adapter = ItemAdapter(item)
        year = adapter['year']
        if year not in self.year_to_exporter:
            xml_file = open(f'{year}.xml', 'wb')
            exporter = XmlItemExporter(xml_file)
            exporter.start_exporting()
            self.year_to_exporter[year] = (exporter, xml_file)
        return self.year_to_exporter[year][0]

    def process_item(self, item, spider):
        exporter = self._exporter_for_item(item)
        exporter.export_item(item)
        return item

```

Serialization of item fields

By default, the field values are passed unmodified to the underlying serialization library, and the decision of how to serialize them is delegated to each particular serialization library.

However, you can customize how each field value is serialized *before it is passed to the serialization library*.

There are two ways to customize how a field will be serialized, which are described next.

1. Declaring a serializer in the field

If you use `:class:`~scrapy.Item`` you can declare a serializer in the `:ref:`field metadata <topics-items-fields>``. The serializer must be a callable which receives a value and returns its serialized form.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 93); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 93); [backlink](#)

Unknown interpreted text role "ref".

Example:

```

import scrapy

def serialize_price(value):
    return f'$ {str(value)}'

class Product(scrapy.Item):
    name = scrapy.Field()
    price = scrapy.Field(serializer=serialize_price)

```

2. Overriding the `serialize_field()` method

You can also override the `:meth:`~BaseItemExporter.serialize_field`` method to customize how your field value will be exported.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 112); [backlink](#)

Unknown interpreted text role "meth".

Make sure you call the base class `:meth:`~BaseItemExporter.serialize_field`` method after your custom code.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 115); [backlink](#)

Unknown interpreted text role "meth".

Example:

```
from scrapy.exporter import XmlItemExporter

class ProductXmlExporter(XmlItemExporter):

    def serialize_field(self, field, name, value):
        if name == 'price':
            return f'$ {str(value)}'
        return super().serialize_field(field, name, value)
```

Built-in Item Exporters reference

Here is a list of the Item Exporters bundled with Scrapy. Some of them contain output examples, which assume you're exporting these two items:

```
Item(name='Color TV', price='1200')
Item(name='DVD player', price='200')
```

BaseItemExporter

This is the (abstract) base class for all Item Exporters. It provides support for common features used by all (concrete) Item Exporters, such as defining what fields to export, whether to export empty fields, or which encoding to use.

These features can be configured through the `__init__` method arguments which populate their respective instance attributes: `attr:'fields_to_export'`, `attr:'export_empty_fields'`, `attr:'encoding'`, `attr:'indent'`.

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

Unknown interpreted text role "attr".

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

Unknown interpreted text role "attr".

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

Unknown interpreted text role "attr".

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

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 154)

Unknown directive type "versionadded".

```
.. versionadded:: 2.0
   The *dont_fail* parameter.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 157)

Unknown directive type "method".

```
.. method:: export_item(item)

   Exports the given item. This method must be implemented in subclasses.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 161)

Unknown directive type "method".

```
.. method:: serialize_field(field, name, value)
```

Return the serialized value for the given field. You can override this method (in your custom Item Exporters) if you want to control how a particular field or value will be serialized/exported.

By default, this method looks for a serializer `:ref:`declared in the item field <topics-exporters-serializers>`` and returns the result of applying that serializer to the value. If no serializer is found, it returns the value unchanged.

`:param field:` the field being serialized. If the source `:ref:`item object <item-types>`` does not define field metadata, `*field*` is an empty `:class:`dict``.

`:type field:` `:class:`~scrapy.Field`` object or a `:class:`dict`` instance

`:param name:` the name of the field being serialized

`:type name:` str

`:param value:` the value being serialized

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 182)

Unknown directive type "method".

```
.. method:: start_exporting()
```

Signal the beginning of the exporting process. Some exporters may use this to generate some required header (for example, the `:class:`XmlItemExporter``). You must call this method before exporting any items.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 189)

Unknown directive type "method".

```
.. method:: finish_exporting()
```

Signal the end of the exporting process. Some exporters may use this to generate some required footer (for example, the `:class:`XmlItemExporter``). You must always call this method after you have no more items to export.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 196)

Unknown directive type "attribute".

```
.. attribute:: fields_to_export
```

A list with the name of the fields that will be exported, or ```None``` if you want to export all fields. Defaults to ```None```.

Some exporters (like `:class:`CsvItemExporter``) respect the order of the fields defined in this attribute.

When using `:ref:`item objects <item-types>`` that do not expose all their possible fields, exporters that do not support exporting a different subset of fields per item will only export the fields found in the first item exported. Use ```fields_to_export``` to define all the fields to be exported.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 210)

Unknown directive type "attribute".

```
.. attribute:: export_empty_fields
```

Whether to include empty/unpopulated item fields in the exported data. Defaults to ``False``. Some exporters (like :class:`CsvItemExporter`) ignore this attribute and always export all empty fields.

This option is ignored for dict items.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 218)

Unknown directive type "attribute".

```
.. attribute:: encoding
```

The output character encoding.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 222)

Unknown directive type "attribute".

```
.. attribute:: indent
```

Amount of spaces used to indent the output on each level. Defaults to ``0``.

- * ``indent=None`` selects the most compact representation, all items in the same line with no indentation
- * ``indent<=0`` each item on its own line, no indentation
- * ``indent>0`` each item on its own line, indented with the provided numeric value

PythonItemExporter

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 234)

Unknown directive type "autoclass".

```
.. autoclass:: PythonItemExporter
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 237)

Unknown directive type "highlight".

```
.. highlight:: none
```

XmlItemExporter

Exports items in XML format to the specified file object.

param file: the file-like object to use for exporting the data. Its `write` method should accept bytes (a disk file opened in binary mode, a `io.BytesIO` object, etc)

param root_element: The name of root element in the exported XML.

type root_element: str

param item_element: The name of each item element in the exported XML.

type item_element: str

The additional keyword arguments of this `__init__` method are passed to the :class:`BaseItemExporter` `__init__` method.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 255); [backlink](#)

Unknown interpreted text role "class".

A typical output of this exporter would be:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<items>
  <item>
    <name>Color TV</name>
    <price>1200</price>
  </item>
  <item>
    <name>DVD player</name>
    <price>200</price>
  </item>
</items>
```

Unless overridden in the `meth:'serialize_field'` method, multi-valued fields are exported by serializing each value inside a `<value>` element. This is for convenience, as multi-valued fields are very common.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 272); [backlink](#)
Unknown interpreted text role "meth".

For example, the item:

```
Item(name=['John', 'Doe'], age='23')
```

Would be serialized as:

```
<?xml version="1.0" encoding="utf-8"?>
<items>
  <item>
    <name>
      <value>John</value>
      <value>Doe</value>
    </name>
    <age>23</age>
  </item>
</items>
```

CsvItemExporter

Exports items in CSV format to the given file-like object. If the `attr:'fields_to_export'` attribute is set, it will be used to define the CSV columns and their order. The `attr:'export_empty_fields'` attribute has no effect on this exporter.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 298); [backlink](#)
Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 298); [backlink](#)
Unknown interpreted text role "attr".

param file: the file-like object to use for exporting the data. Its `write` method should accept bytes (a disk file opened in binary mode, a `io.BytesIO` object, etc)

param include_headers_line: If enabled, makes the exporter output a header line with the field names taken from `attr:'BaseItemExporter.fields_to_export'` or the first exported item fields.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 306); [backlink](#)
Unknown interpreted text role "attr".

type include_headers_line:
bool

param join_multivalued: The char (or chars) that will be used for joining multi-valued fields, if found.

type include_headers_line:
str

param errors:

The optional string that specifies how encoding and decoding errors are to be handled. For more information see `class:'io.TextIOWrapper'`.

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

Unknown interpreted text role "class".

type errors: str

The additional keyword arguments of this `__init__` method are passed to the `class:'BaseItemExporter'` `__init__` method, and the leftover arguments to the `func:'csv.writer'` function, so you can use any `func:'csv.writer'` function argument to customize this exporter.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 320); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 320); [backlink](#)

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 320); [backlink](#)

Unknown interpreted text role "func".

A typical output of this exporter would be:

```
product,price
Color TV,1200
DVD player,200
```

PickleItemExporter

Exports items in pickle format to the given file-like object.

param file: the file-like object to use for exporting the data. Its `write` method should accept `bytes` (a disk file opened in binary mode, a `io.BytesIO` object, etc)

param protocol: The pickle protocol to use.

type protocol: int

For more information, see `mod:'pickle'`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 344); [backlink](#)

Unknown interpreted text role "mod".

The additional keyword arguments of this `__init__` method are passed to the `class:'BaseItemExporter'` `__init__` method.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 346); [backlink](#)

Unknown interpreted text role "class".

Pickle isn't a human readable format, so no output examples are provided.

PprintItemExporter

Exports items in pretty print format to the specified file object.

param file: the file-like object to use for exporting the data. Its `write` method should accept `bytes` (a disk file opened in binary mode, a `io.BytesIO` object, etc)

The additional keyword arguments of this `__init__` method are passed to the `class:'BaseItemExporter'` `__init__` method.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 361); [backlink](#)

Unknown interpreted text role "class".

A typical output of this exporter would be:

```
{'name': 'Color TV', 'price': '1200'}  
{'name': 'DVD player', 'price': '200'}
```

Longer lines (when present) are pretty-formatted.

JsonItemExporter

Exports items in JSON format to the specified file-like object, writing all objects as a list of objects. The additional `__init__` method arguments are passed to the `:class:`BaseItemExporter`` `__init__` method, and the leftover arguments to the `:class:`~json.JSONEncoder`` `__init__` method, so you can use any `:class:`~json.JSONEncoder`` `__init__` method argument to customize this exporter.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 376); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 376); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 376); [backlink](#)

Unknown interpreted text role "class".

param file: the file-like object to use for exporting the data. Its `write` method should accept bytes (a disk file opened in binary mode, a `io.BytesIO` object, etc)

A typical output of this exporter would be:

```
[{"name": "Color TV", "price": "1200"},  
{"name": "DVD player", "price": "200"}]
```

Warning

JSON is very simple and flexible serialization format, but it doesn't scale well for large amounts of data since incremental (aka. stream-mode) parsing is not well supported (if at all) among JSON parsers (on any language), and most of them just parse the entire object in memory. If you want the power and simplicity of JSON with a more stream-friendly format, consider using `:class:`JsonLinesItemExporter`` instead, or splitting the output in multiple chunks.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 392); [backlink](#)

Unknown interpreted text role "class".

JsonLinesItemExporter

Exports items in JSON format to the specified file-like object, writing one JSON-encoded item per line. The additional `__init__` method arguments are passed to the `:class:`BaseItemExporter`` `__init__` method, and the leftover arguments to the `:class:`~json.JSONEncoder`` `__init__` method, so you can use any `:class:`~json.JSONEncoder`` `__init__` method argument to customize this exporter.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) exporters.rst, line 405); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 405); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 405); [backlink](#)

Unknown interpreted text role "class".

param file: the file-like object to use for exporting the data. Its `write` method should accept bytes (a disk file opened in binary mode, a `io.BytesIO` object, etc)

A typical output of this exporter would be:

```
{"name": "Color TV", "price": "1200"}
{"name": "DVD player", "price": "200"}
```

Unlike the one produced by `:class:`JsonItemExporter``, the format produced by this exporter is well suited for serializing large amounts of data.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 419); [backlink](#)

Unknown interpreted text role "class".

MarshalItemExporter

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\scrapy-master) (docs) (topics) exporters.rst, line 425)

Unknown directive type "autoclass".

```
.. autoclass:: MarshalItemExporter
```