Stats Collection

Scrapy provides a convenient facility for collecting stats in the form of key/values, where values are often counters. The facility is called the Stats Collector, and can be accessed through the :attr:`~scrapy.crawler.Crawler.stats` attribute of the :ref:`topics-api-crawler`, as illustrated by the examples in the :ref:`topics-stats-usecases` section below.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)stats.rst`, **line 7**); *backlink*
>
> Unknown interpreted text role "attr".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)stats.rst`, **line 7**); *backlink*
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)stats.rst`, **line 7**); *backlink*
>
> Unknown interpreted text role "ref".

However, the Stats Collector is always available, so you can always import it in your module and use its API (to increment or set new stat keys), regardless of whether the stats collection is enabled or not. If it's disabled, the API will still work but it won't collect anything. This is aimed at simplifying the stats collector usage: you should spend no more than one line of code for collecting stats in your spider, Scrapy extension, or whatever code you're using the Stats Collector from.

Another feature of the Stats Collector is that it's very efficient (when enabled) and extremely efficient (almost unnoticeable) when disabled.

The Stats Collector keeps a stats table per open spider which is automatically opened when the spider is opened, and closed when the spider is closed.

Common Stats Collector uses

Access the stats collector through the :attr:`~scrapy.crawler.Crawler.stats` attribute. Here is an example of an extension that access stats:

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)stats.rst`, **line 32**); *backlink*
>
> Unknown interpreted text role "attr".

```python
class ExtensionThatAccessStats:

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

    @classmethod
    def from_crawler(cls, crawler):
        return cls(crawler.stats)
```

Set stat value:

```python
stats.set_value('hostname', socket.gethostname())
```

Increment stat value:

```python
stats.inc_value('custom_count')
```

Set stat value only if greater than previous:

```python
stats.max_value('max_items_scraped', value)
```

Set stat value only if lower than previous:

```python
stats.min_value('min_free_memory_percent', value)
```

Get stat value:

```python
>>> stats.get_value('custom_count')
```

```
1
```

Get all stats:

```
>>> stats.get_stats()
{'custom_count': 1, 'start_time': datetime.datetime(2009, 7, 14, 21, 47, 28, 977139)}
```

Available Stats Collectors

Besides the basic :class:`StatsCollector` there are other Stats Collectors available in Scrapy which extend the basic Stats Collector. You can select which Stats Collector to use through the :setting:`STATS_CLASS` setting. The default Stats Collector used is the :class:`MemoryStatsCollector`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)stats.rst`, **line 73**); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)stats.rst`, **line 73**); *backlink*
>
> Unknown interpreted text role "setting".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)stats.rst`, **line 73**); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)stats.rst`, **line 78**)
>
> Unknown directive type "currentmodule".
>
> ```
>     .. currentmodule:: scrapy.statscollectors
> ```

MemoryStatsCollector

A simple stats collector that keeps the stats of the last scraping run (for each spider) in memory, after they're closed. The stats can be accessed through the :attr:`spider_stats` attribute, which is a dict keyed by spider domain name.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)stats.rst`, **line 85**); *backlink*
>
> Unknown interpreted text role "attr".

This is the default Stats Collector used in Scrapy.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)stats.rst`, **line 92**)
>
> Unknown directive type "attribute".
>
> ```
>     .. attribute:: spider_stats
>
>       A dict of dicts (keyed by spider name) containing the stats of the last
>       scraping run for each spider.
> ```

DummyStatsCollector

A Stats collector which does nothing but is very efficient (because it does nothing). This stats collector can be set via the :setting:`STATS_CLASS` setting, to disable stats collect in order to improve performance. However, the performance penalty of stats collection is usually marginal compared to other Scrapy workload like parsing pages.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)stats.rst`, **line 102**); *backlink*
>
> Unknown interpreted text role "setting".