Debugging Spiders

This document explains the most common techniques for debugging spiders. Consider the following Scrapy spider below:

```
import scrapy
from myproject.items import MyItem

class MySpider(scrapy.Spider):
    name = 'myspider'
    start_urls = (
        'http://example.com/page1',
        'http://example.com/page2',
        )

    def parse(self, response):
        # <processing code not shown>
        # collect `item_urls`
        for item_url in item_urls:
            yield scrapy.Request(item_url, self.parse_item)

    def parse_item(self, response):
        # <processing code not shown>
        item = MyItem()
        # populate `item` fields
        # and extract item_details_url
        yield scrapy.Request(item_details_url, self.parse_details, cb_kwargs={'item': item})

    def parse_details(self, response, item):
        # populate more `item` fields
        return item
```

Basically this is a simple spider which parses two pages of items (the start_urls). Items also have a details page with additional information, so we use the `cb_kwargs` functionality of :class:`~scrapy.Request` to pass a partially populated item.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)debug.rst`, line 37); *backlink*

Unknown interpreted text role "class".

Parse Command

The most basic way of checking the output of your spider is to use the :command:`parse` command. It allows to check the behaviour of different parts of the spider at the method level. It has the advantage of being flexible and simple to use, but does not allow debugging code inside a method.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)debug.rst`, line 46); *backlink*

Unknown interpreted text role "command".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)debug.rst`, line 51)

Unknown directive type "highlight".

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

In order to see the item scraped from a specific url:

```
$ scrapy parse --spider=myspider -c parse_item -d 2 <item_url>
[ ... scrapy log lines crawling example.com spider ... ]

>>> STATUS DEPTH LEVEL 2 <<<
# Scraped Items  ------------------------------------------------------------
[{'url': <item_url>}]

# Requests  -----------------------------------------------------------------
[]
```

Using the `--verbose` or `-v` option we can see the status at each depth level:

```
$ scrapy parse --spider=myspider -c parse_item -d 2 -v <item_url>
```

```
[ ... scrapy log lines crawling example.com spider ... ]

>>> DEPTH LEVEL: 1 <<<
# Scraped Items  ------------------------------------------------------------
[]

# Requests  -----------------------------------------------------------------
[<GET item_details_url>]


>>> DEPTH LEVEL: 2 <<<
# Scraped Items  ------------------------------------------------------------
[{'url': <item_url>}]

# Requests  -----------------------------------------------------------------
[]
```

Checking items scraped from a single start_url, can also be easily achieved using:

```
$ scrapy parse --spider=myspider -d 3 'http://example.com/page1'
```

Scrapy Shell

While the :command:`parse` command is very useful for checking behaviour of a spider, it is of little help to check what happens inside a callback, besides showing the response received and the output. How to debug the situation when `parse_details` sometimes receives no item?

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master) (docs) (topics)debug.rst`, **line 98**); ***backlink***

Unknown interpreted text role "command".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master) (docs) (topics)debug.rst`, **line 103**)

Unknown directive type "highlight".

```
.. highlight:: python
```

Fortunately, the :command:`shell` is your bread and butter in this case (see :ref:`topics-shell-inspect-response`):

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master) (docs) (topics)debug.rst`, **line 105**); ***backlink***

Unknown interpreted text role "command".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master) (docs) (topics)debug.rst`, **line 105**); ***backlink***

Unknown interpreted text role "ref".

```
from scrapy.shell import inspect_response

def parse_details(self, response, item=None):
    if item:
        # populate more `item` fields
        return item
    else:
        inspect_response(response, self)
```

See also: :ref:`topics-shell-inspect-response`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master) (docs) (topics)debug.rst`, **line 117**); ***backlink***

Unknown interpreted text role "ref".

Open in browser

Sometimes you just want to see how a certain response looks in a browser, you can use the `open_in_browser` function for that. Here is an example of how you would use it:

```
from scrapy.utils.response import open_in_browser

def parse_details(self, response):
    if "item name" not in response.body:
        open_in_browser(response)
```

`open_in_browser` will open a browser with the response received by Scrapy at that point, adjusting the base tag so that images and styles are displayed properly.

Logging

Logging is another useful option for getting information about your spider run. Although not as convenient, it comes with the advantage that the logs will be available in all future runs should they be necessary again:

```
def parse_details(self, response, item=None):
    if item:
        # populate more `item` fields
        return item
    else:
        self.logger.warning('No item received for %s', response.url)
```

For more information, check the :ref:`topics-logging` section.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\(scrapy-master)(docs)(topics)debug.rst`, line 150); *backlink*

Unknown interpreted text role "ref".