Spiders Contracts

Testing spiders can get particularly annoying and while nothing prevents you from writing unit tests the task gets cumbersome quickly. Scrapy offers an integrated way of testing your spiders by the means of contracts.

This allows you to test each callback of your spider by hardcoding a sample url and check various constraints for how the callback processes the response. Each contract is prefixed with an `@` and included in the docstring. See the following example:

```
def parse(self, response):
    """ This function parses a sample response. Some contracts are mingled
    with this docstring.

    @url http://www.amazon.com/s?field-keywords=selfish+gene
    @returns items 1 16
    @returns requests 0 0
    @scrapes Title Author Year Price
    """
```

This callback is tested using three built-in contracts:

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]contracts.rst, line 28)**
>
> Unknown directive type "module".
>
> ```
> .. module:: scrapy.contracts.default
> ```

This contract (`@url`) sets the sample URL used when checking other contract conditions for this spider. This contract is mandatory. All callbacks lacking this contract are ignored when running the checks:

```
@url url
```

This contract (`@cb_kwargs`) sets the :attr:`cb_kwargs <scrapy.Request.cb_kwargs>` attribute for the sample request. It must be a valid JSON dictionary.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]contracts.rst, line 40); *backlink***
>
> Unknown interpreted text role "attr".

```
@cb_kwargs {"arg1": "value1", "arg2": "value2", ...}
```

This contract (`@returns`) sets lower and upper bounds for the items and requests returned by the spider. The upper bound is optional:

```
@returns item(s)|request(s) [min [max]]
```

This contract (`@scrapes`) checks that all the items returned by the callback have the specified fields:

```
@scrapes field_1 field_2 ...
```

Use the :command:`check` command to run the contract checks.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]contracts.rst, line 60); *backlink***
>
> Unknown interpreted text role "command".

Custom Contracts

If you find you need more power than the built-in Scrapy contracts you can create and load your own contracts in the project by using the :setting:`SPIDER_CONTRACTS` setting:

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]contracts.rst, line 65); *backlink***
>
> Unknown interpreted text role "setting".

```
SPIDER_CONTRACTS = {
    'myproject.contracts.ResponseCheck': 10,
```

```
    'myproject.contracts.ItemValidate': 10,
}
```

Each contract must inherit from :class:`~scrapy.contracts.Contract` and can override three methods:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]contracts.rst, line 74); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]contracts.rst, line 77)

Unknown directive type "module".

```
.. module:: scrapy.contracts
```

param method: callback function to which the contract is associated
type method: collections.abc.Callable
param args: list of arguments passed into the docstring (whitespace separated)
type args: list

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]contracts.rst, line 88)

Unknown directive type "method".

```
.. method:: Contract.adjust_request_args(args)

    This receives a ``dict`` as an argument containing default arguments
    for request object. :class:`~scrapy.Request` is used by default,
    but this can be changed with the ``request_cls`` attribute.
    If multiple contracts in chain have this attribute defined, the last one is used.

    Must return the same or a modified version of it.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]contracts.rst, line 97)

Unknown directive type "method".

```
.. method:: Contract.pre_process(response)

    This allows hooking in various checks on the response received from the
    sample request, before it's being passed to the callback.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]contracts.rst, line 102)

Unknown directive type "method".

```
.. method:: Contract.post_process(output)

    This allows processing the output of the callback. Iterators are
    converted listified before being passed to this hook.
```

Raise :class:`~scrapy.exceptions.ContractFail` from :class:`~scrapy.contracts.Contract.pre_process` or :class:`~scrapy.contracts.Contract.post_process` if expectations are not met:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]contracts.rst, line 107); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]contracts.rst, line 107); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]contracts.rst, line 107); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]contracts.rst, line 111)

Unknown directive type "autoclass".

```
.. autoclass:: scrapy.exceptions.ContractFail
```

Here is a demo contract which checks the presence of a custom header in the response received:

```
from scrapy.contracts import Contract
from scrapy.exceptions import ContractFail

class HasHeaderContract(Contract):
    """ Demo contract which checks the presence of a custom header
        @has_header X-CustomHeader
    """

    name = 'has_header'

    def pre_process(self, response):
        for header in self.args:
            if header not in response.headers:
                raise ContractFail('X-CustomHeader not present')
```

Detecting check runs

When `scrapy check` is running, the `SCRAPY_CHECK` environment variable is set to the `true` string. You can use :data:`os.environ` to perform any change to your spiders or your settings when `scrapy check` is used:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\[scrapy-master][docs][topics]contracts.rst, line 136); *backlink*

Unknown interpreted text role "data".

```
import os
import scrapy

class ExampleSpider(scrapy.Spider):
    name = 'example'

    def __init__(self):
        if os.environ.get('SCRAPY_CHECK'):
            pass  # Do some scraper adjustments when a check is running
```