SEP	1
Title	API for populating item fields (comparison)
Author	Ismael Carnales, Pablo Hoffman, Daniel Grana
Created	2009-07-19
Status	Obsoleted by :ref:`sep-008`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\sep\[scrapy-master][sep]sep-001.rst, line 7); *backlink*

Unknown interpreted text role "ref".

SEP-001 - API for populating item fields (comparison)

This page shows different usage scenarios for the two new proposed API for populating item field values (which will replace the old deprecated !RobustItem API) and compares them. One of these will be chosen as the recommended (and supported) mechanism in Scrapy 0.7.

Candidates and their API

RobustItem (old, deprecated)

- `attribute(field_name, selector_or_value, **modifiers_and_adaptor_args)`

> **Note**
>
> `attribute()` modifiers (like `add=True`) are passed together with adaptor args as keyword arguments (this is ugly)

ItemForm

- `__init__(response, item=None, **adaptor_args)`
  - instantiate an `ItemForm` with a item instance with predefined adaptor arguments
- `__setitem__(field_name, selector_or_value)`
  - set field value
- `__getitem__(field_name)`
  - return the "computed" value of a field (the one that would be set to the item). returns `None` if not set.
- `get_item()` - return the item populated with the data provided so far

ItemBuilder

- `__init__(response, item=None, **adaptor_args)`
  - instantiate an `ItemBuilder` with predefined adaptor arguments
- `add_value(field_name, selector_or_value, **adaptor_args)`
  - add value to field
- `replace_value(field_name, selector_or_value, **adaptor_args)`
  - replace existing field value
- `get_value(field_name)`
  - return the "computed" value of a field (the one that would be set to the item). returns `None` if not set.
- `get_item()`
  - return the item populated with the data provided so far

Pros and cons of each candidate

ItemForm

Pros: - same API used for Items (see https://docs.scrapy.org/en/latest/topics/items.html) - some people consider setitem API more elegant than methods API

Cons: - doesn't allow passing run-time arguments to adaptors on assign, you have to

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\sep\[scrapy-master][sep]sep-001.rst, line 69)

Unexpected indentation.

override the adaptors for your spider if you need specific parameters, which can be an overhead. Example:

Neutral: - solves the add=True problem using standard `__add__` and `list.append()` method

ItemBuilder

Pros: - allows passing run-time arguments to adaptors on assigned

Cons: - some people consider setitem API more elegant than methods API

Neutral: - solves the "add=True" problem by implementing different methods per action

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\sep\[scrapy-master][sep]sep-001.rst, line 86)

Unexpected indentation.

(replacing or adding)

Usage Scenarios for each candidate

Defining adaptors

ItemForm

```
#!python
class NewsForm(ItemForm):
    item_class = NewsItem

    url = adaptor(extract, remove_tags(), unquote(), strip)
    headline = adaptor(extract, remove_tags(), unquote(), strip)
```

ItemBuilder

```
#!python
class NewsBuilder(ItemBuilder):
    item_class = NewsItem

    url = adaptor(extract, remove_tags(), unquote(), strip)
    headline = adaptor(extract, remove_tags(), unquote(), strip)
```

Creating an Item

ItemForm

```
#!python
ia = NewsForm(response)
ia['url'] = response.url
ia['headline'] = x.x('//h1[@class="headline"]')

# if we want to add another value to the same field
ia['headline'] += x.x('//h1[@class="headline2"]')

# if we want to replace the field value other value to the same field
ia['headline'] = x.x('//h1[@class="headline3"]')

return ia.get_item()
```

ItemBuilder

```
#!python
il = NewsBuilder(response)
il.add_value('url', response.url)
il.add_value('headline', x.x('//h1[@class="headline"]'))

# if we want to add another value to the same field
il.add_value('headline', x.x('//h1[@class="headline2"]'))

# if we want to replace the field value other value to the same field
il.replace_value('headline', x.x('//h1[@class="headline3"]'))

return il.get_item()
```

Using different adaptors per Spider/Site

ItemForm

```
#!python
class SiteNewsFrom(NewsForm):
    published = adaptor(HtmlNewsForm.published, to_date('%d.%m.%Y'))
```

ItemBuilder

```
#!python
class SiteNewsBuilder(NewsBuilder):
    published = adaptor(HtmlNewsBuilder.published, to_date('%d.%m.%Y'))
```

Check the value of an item being-extracted

ItemForm

```
#!python
ia = NewsForm(response)
ia['headline'] = x.x('//h1[@class="headline"]')
if not ia['headline']:
    ia['headline'] = x.x('//h1[@class="title"]')
```

ItemBuilder

```
#!python
il = NewsBuilder(response)
il.add_value('headline', x.x('//h1[@class="headline"]'))
if not nf.get_value('headline'):
    il.add_value('headline', x.x('//h1[@class="title"]'))
```

Adding a value to a list attribute/field

ItemForm

```
#!python
ia['headline'] += x.x('//h1[@class="headline"]')
```

ItemBuilder

```
#!python
il.add_value('headline', x.x('//h1[@class="headline"]'))
```

Passing run-time arguments to adaptors

ItemForm

```
#!python
# Only approach is passing arguments when instantiating the form
ia = NewsForm(response, default_unit='cm')
ia['width'] = x.x('//p[@class="width"]')
```

ItemBuilder

```
#!python
il.add_value('width', x.x('//p[@class="width"]'), default_unit='cm')

# an alternative approach (more efficient)
il = NewsBuilder(response, default_unit='cm')
il.add_value('width', x.x('//p[@class="width"]'))
```

Passing run-time arguments to adaptors (same argument name)

ItemForm

```
#!python
class MySiteForm(ItemForm):
    width = adaptor(ItemForm.width, default_unit='cm')
    volume = adaptor(ItemForm.width, default_unit='lt')

ia['width'] = x.x('//p[@class="width"]')
ia['volume'] = x.x('//p[@class="volume"]')

# another example passing parameters on instance
ia = NewsForm(response, encoding='utf-8')
ia['name'] = x.x('//p[@class="name"]')
```

ItemBuilder

```
#!python
il.add_value('width', x.x('//p[@class="width"]'), default_unit='cm')
il.add_value('volume', x.x('//p[@class="volume"]'), default_unit='lt')
```