SEP	8
Title	Item Parsers
Author	Pablo Hoffman
Created	2009-08-11
Status	Final (implemented with variations)
Obsoletes	:doc:`sep-001`, :doc:`sep-002`, :doc:`sep-003`, :doc:`sep-005` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\sep\[scrapy-master][sep]sep-008.rst`, line 8); *backlink* Unknown interpreted text role "doc". **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\sep\[scrapy-master][sep]sep-008.rst`, line 8); *backlink* Unknown interpreted text role "doc". **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\sep\[scrapy-master][sep]sep-008.rst`, line 8); *backlink* Unknown interpreted text role "doc". **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\sep\[scrapy-master][sep]sep-008.rst`, line 8); *backlink* Unknown interpreted text role "doc".

SEP-008 - Item Loaders

Item Parser is the final API proposed to implement Item Builders/Loader proposed in :doc:`sep-001`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\sep\[scrapy-master][sep]sep-008.rst`, line 14); *backlink*

Unknown interpreted text role "doc".

Note

This is the API that was finally implemented with the name "Item Loaders", instead of "Item Parsers" along with some other minor fine tuning to the API methods and semantics.

Dataflow

1. `ItemParser.add_value()` 1. **input_parser** 2. store
2. `ItemParser.add_xpath()` *(only available in XPathItemLoader)* 1. selector.extract() 2. **input_parser** 3. store
3. `ItemParser.populate_item()` *(ex. get_item)* 1. **output_parser** 2. assign field

Modules and classes

- `scrapy.contrib.itemparser.ItemParser`
- `scrapy.contrib.itemparser.XPathItemParser`
- `scrapy.contrib.itemparser.parsers.``MapConcat` *(ex. ``TreeExpander``)*
- `scrapy.contrib.itemparser.parsers.``TakeFirst`
- `scrapy.contrib.itemparser.parsers.Join`
- `scrapy.contrib.itemparser.parsers.Identity`

Public API

- `ItemParser.add_value()`
- `ItemParser.replace_value()`
- `ItemParser.populate_item()` *(returns item populated)*
- `ItemParser.get_collected_values()` *(note the 's' in values)*
- `ItemParser.parse_field()`
- `ItemParser.get_input_parser()`

- `ItemParser.get_output_parser()`
- `ItemParser.context`
- `ItemParser.default_item_class`
- `ItemParser.default_input_parser`
- `ItemParser.default_output_parser`
- `ItemParser.*field*_in`
- `ItemParser.*field*_out`

Alternative Public API Proposal

- `ItemLoader.add_value()`
- `ItemLoader.replace_value()`
- `ItemLoader.load_item()` *(returns loaded item)*
- `ItemLoader.get_stored_values()` or `ItemLoader.get_values()` *(returns the ``ItemLoader values)*
- `ItemLoader.get_output_value()`
- `ItemLoader.get_input_processor()` or `ItemLoader.get_in_processor()` *(short version)*
- `ItemLoader.get_output_processor()` or `ItemLoader.get_out_processor()` *(short version)*
- `ItemLoader.context`
- `ItemLoader.default_item_class`
- `ItemLoader.default_input_processor` or `ItemLoader.default_in_processor` *(short version)*
- `ItemLoader.default_output_processor` or `ItemLoader.default_out_processor` *(short version)*
- `ItemLoader.*field*_in`
- `ItemLoader.*field*_out`

Usage example: declaring Item Parsers

```
#!python
from scrapy.contrib.itemparser import XPathItemParser, parsers

class ProductParser(XPathItemParser):
    name_in = parsers.MapConcat(removetags, filterx)
    price_in = parsers.MapConcat(...)

    price_out = parsers.TakeFirst()
```

Usage example: declaring parsers in Fields

```
#!python
class Product(Item):
    name = Field(output_parser=parsers.Join(), ...)
    price = Field(output_parser=parsers.TakeFirst(), ...)

    description = Field(input_parser=parsers.MapConcat(removetags))
```