
autoextract-poet Documentation

Release 0.4.0

Zyte

Mar 09, 2022

GETTING STARTED

1	Introduction	3
2	API Reference	5
3	Contributing	53
4	Changelog	55
5	License	57
	Python Module Index	59
	Index	61

autoextract-poet contains the common item definitions. Such items can be extracted automatically using Zyte [AutoExtract API](#) (you can use [scrapy-poet](#) and [scrapy-autoextract](#) for this).

The [AutoExtract API](#) is able to convert pages into data automatically. It support multiple types of pages, like articles, products, real estate, comments, job posting, reviews, etc. See the full list of supported page types [here](#).

See also [web-poet](#) for an introduction about the Page Objects paradigm and the [scrapy-poet tutorial](#) for an introduction about how to use Page Objects with Scrapy spiders.

License is BSD 3-clause.

INTRODUCTION

1.1 Installing autoextract-poet

`autoextract-poet` is a regular PyPI package that can be installed using `pip`: `pip install autoextract-poet`. It is also a dependency of `scrapy-autoextract`, and installed automatically if you use `scrapy-autoextract`.

1.2 Basic usage

You can use items defined by `autoextract-poet` just as regular Python objects, to standardize item definitions. They are implemented as `attrs` classes, and can be used as `Scrapy` items directly, or converted to dictionaries (e.g. for serialization) via `itemadapter`. The full list of items can be seen here [autoextract_poet.items](#).

`scrapy-autoextract` provides an automatic way to extract items defined here from any website, using `Scrapy` and `Autoextract API`. See its [scrapy-autoextract documentation](#) for more.

1.3 Compatibility with new fields added to the API

Eventually, some new fields could be added to the `Autoextract API`. When you're creating `autoextract-poet` items from `Autoextract` responses, the library would ignore unknown fields by default, until you upgrade the library to a version containing the new field. But you might want to keep the unknown (new) fields even if you don't update the `autoextract-poet` library.

If you're using `Scrapy` (or `itemadapter`), you can make these unknown attributes exposed in the output by registering `AutoExtractAdapter` in `itemadapter`'s `ADAPTER_CLASSES`:

```
from autoextract_poet import AutoExtractAdapter
from itemadapter import ItemAdapter
ItemAdapter.ADAPTER_CLASSES.appendleft(AutoExtractAdapter)
```

For example, you can put this code to `settings.py` of your `Scrapy` project.

API REFERENCE

autoextract_poet

2.1 autoextract_poet

Modules

autoextract_poet.adapters

autoextract_poet.items

autoextract_poet.page_inputs

autoextract_poet.pages

autoextract_poet.util

2.1.1 autoextract_poet.adapters

Classes

AutoExtractAdapter(item)

ItemAdapter for AutoExtract poet items that deals transparently with both the `attr.s` defined fields and the rest of unknown fields received at item initialization, offering an unified view over all the item fields.

autoextract_poet.adapters.AutoExtractAdapter

class AutoExtractAdapter(*item: Any*)

Bases: `itemadapter.adapter.AttrsAdapter`

`ItemAdapter` for `AutoExtract` poet items that deals transparently with both the `attr.s` defined fields and the rest of unknown fields received at item initialization, offering an unified view over all the item fields.

The utility is twofold. Firstly, it serves to ensure the pass-through of new fields from the API even if `autoextract-poet` item definitions have not been yet updated. In other words, it can be used to create spiders that preserve all the data coming from the API even if they don't have updated item definitions.

Secondly, it offers a common interface to access and modify both kind of fields (known and unknown).

Remember that this adapter should be enabled by invoking:

```
ItemAdapter.ADAPTER_CLASSES.appendleft(AutoExtractAdapter)
```

`__init__`(*item: Any*) → `None`

Methods

<code>__init__</code> (<i>item</i>)	
<code>clear</code> ()	
<code>field_names</code> ()	Return a dynamic view of the item's field names.
<code>get</code> (<i>k</i> , <i>d</i>)	
<code>get_field_meta</code> (<i>field_name</i>)	Return metadata for the given field name, if available.
<code>get_field_meta_from_class</code> (<i>item_class</i> , <i>field_name</i>)	
<code>is_item</code> (<i>item</i>)	Return True if the adapter can handle the given item, False otherwise.
<code>is_item_class</code> (<i>item_class</i>)	Return True if the adapter can handle the given item class, False otherwise.
<code>items</code> ()	
<code>keys</code> ()	
<code>pop</code> (<i>k</i> , <i>d</i>)	If key is not found, <i>d</i> is returned if given, otherwise <code>KeyError</code> is raised.
<code>popitem</code> ()	as a 2-tuple; but raise <code>KeyError</code> if <i>D</i> is empty.
<code>setdefault</code> (<i>k</i> , <i>d</i>)	
<code>update</code> ([<i>E</i> ,]** <i>F</i>)	If <i>E</i> present and has a <code>.keys()</code> method, does: for <i>k</i> in <i>E</i> : <i>D</i> [<i>k</i>] = <i>E</i> [<i>k</i>] If <i>E</i> present and lacks <code>.keys()</code> method, does: for (<i>k</i> , <i>v</i>) in <i>E</i> : <i>D</i> [<i>k</i>] = <i>v</i> In either case, this is followed by: for <i>k</i> , <i>v</i> in <i>F</i> .items(): <i>D</i> [<i>k</i>] = <i>v</i>
<code>values</code> ()	

Attributes

field_names() → `KeysView`

Return a dynamic view of the item's field names.

get_field_meta(*field_name*: `str`) → `mappingproxy`

Return metadata for the given field name, if available.

classmethod is_item(*item*: `Any`) → `bool`

Return True if the adapter can handle the given item, False otherwise.

2.1.2 autoextract_poet.items

Classes

AdditionalProperty(name[, value])

Address([postalCode, streetAddress, ...])

Area([value, unitCode, raw])

Article([url, probability, headline, ...])

ArticleFromList([url, probability, ...])

ArticleList([url, articles, paginationNext, ...])

AvailableAtOrFrom([raw])

Breadcrumb([name, link])

Comment([probability, text, datePublished, ...])

Comments([url, comments])

ForumPost([probability, text, ...])

ForumPosts([url, topic, posts])

FuelEfficiency([raw])

GTIN(type, value)

Item()

JobPosting([probability, url, title, ...])

Location([raw])

continues on next page

Table 6 – continued from previous page

<i>MileageFromOdometer</i> ([value, unitCode])
<i>Offer</i> ([price, currency, availability, ...])
<i>Organization</i> ([raw])
<i>PaginationLink</i> ([url, text])
<i>Product</i> ([url, canonicalUrl, probability, ...])
<i>ProductFromList</i> ([probability, url, name, ...])
<i>ProductList</i> ([url, products, breadcrumbs, ...])
<i>Rating</i> ([ratingValue, bestRating, reviewCount])
<i>RealEstate</i> ([url, probability, name, ...])
<i>Review</i> ([name, reviewBody, reviewRating, ...])
<i>Reviews</i> ([url, reviews, paginationNext, ...])
<i>Salary</i> ([raw, value, currency])
<i>Topic</i> ([name])
<i>TradeAction</i> ([tradeType, price, currency])
<i>Vehicle</i> ([url, canonicalUrl, probability, ...])
<i>VehicleEngine</i> ([raw])

autoextract_poet.items.AdditionalProperty

class `AdditionalProperty`(name: *str*, value: *Optional[str]* = None)

Bases: `autoextract_poet.items.Item`

__init__(name: *str*, value: *Optional[str]* = None) → None

Method generated by attrs for class AdditionalProperty.

Methods

<code>__init__</code> (name[, value])	Method generated by attrs for class AdditionalProperty.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

name

value

autoextract_poet.items.Address

class **Address**(postalCode: *Optional[str] = None*, streetAddress: *Optional[str] = None*, addressCountry: *Optional[str] = None*, addressLocality: *Optional[str] = None*, addressRegion: *Optional[str] = None*, raw: *Optional[str] = None*)

Bases: `autoextract_poet.items.Item`

__init__(postalCode: *Optional[str] = None*, streetAddress: *Optional[str] = None*, addressCountry: *Optional[str] = None*, addressLocality: *Optional[str] = None*, addressRegion: *Optional[str] = None*, raw: *Optional[str] = None*) → None

Method generated by attrs for class Address.

Methods

<code>__init__</code> ([postalCode, streetAddress, ...])	Method generated by attrs for class Address.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

postalCode

streetAddress

addressCountry

addressLocality

addressRegion

raw

autoextract_poet.items.Area

class Area(value: *Optional[float] = None*, unitCode: *Optional[str] = None*, raw: *Optional[str] = None*)

Bases: `autoextract_poet.items.Item`

__init__(value: *Optional[float] = None*, unitCode: *Optional[str] = None*, raw: *Optional[str] = None*) → None

Method generated by attrs for class Area.

Methods

<code>__init__</code> ([value, unitCode, raw])	Method generated by attrs for class Area.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

value
unitCode
raw

autoextract_poet.items.Article

class Article(url: *Optional[str] = None*, probability: *Optional[float] = None*, headline: *Optional[str] = None*, datePublished: *Optional[str] = None*, datePublishedRaw: *Optional[str] = None*, dateModified: *Optional[str] = None*, dateModifiedRaw: *Optional[str] = None*, author: *Optional[str] = None*, authorsList: *List[str] = NOTHING*, inLanguage: *Optional[str] = None*, breadcrumbs: *List[autoextract_poet.items.Breadcrumb] = NOTHING*, mainImage: *Optional[str] = None*, images: *List[str] = NOTHING*, description: *Optional[str] = None*, articleBody: *Optional[str] = None*, articleBodyHtml: *Optional[str] = None*, articleBodyRaw: *Optional[str] = None*, videoUrls: *List[str] = NOTHING*, audioUrls: *List[str] = NOTHING*, canonicalUrl: *Optional[str] = None*)

Bases: `autoextract_poet.items.Item`

__init__(url: *Optional[str] = None*, probability: *Optional[float] = None*, headline: *Optional[str] = None*, datePublished: *Optional[str] = None*, datePublishedRaw: *Optional[str] = None*, dateModified: *Optional[str] = None*, dateModifiedRaw: *Optional[str] = None*, author: *Optional[str] = None*, authorsList: *List[str] = NOTHING*, inLanguage: *Optional[str] = None*, breadcrumbs: *List[autoextract_poet.items.Breadcrumb] = NOTHING*, mainImage: *Optional[str] = None*, images: *List[str] = NOTHING*, description: *Optional[str] = None*, articleBody: *Optional[str] = None*, articleBodyHtml: *Optional[str] = None*, articleBodyRaw: *Optional[str] = None*, videoUrls: *List[str] = NOTHING*, audioUrls: *List[str] = NOTHING*, canonicalUrl: *Optional[str] = None*) → None

Method generated by attrs for class Article.

Methods

<code>__init__</code> (url, probability, headline, ...)	Method generated by attrs for class Article.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

<code>url</code>
<code>probability</code>
<code>headline</code>
<code>datePublished</code>
<code>datePublishedRaw</code>
<code>dateModified</code>
<code>dateModifiedRaw</code>
<code>author</code>
<code>authorsList</code>
<code>inLanguage</code>
<code>breadcrumbs</code>
<code>mainImage</code>
<code>images</code>
<code>description</code>
<code>articleBody</code>
<code>articleBodyHtml</code>
<code>articleBodyRaw</code>
<code>videoUrls</code>
<code>audioUrls</code>
<code>canonicalUrl</code>

classmethod `from_dict(item: Optional[Dict])`

Read an item from a dictionary.

Unknown attributes are kept in the dict `_unknown_fields_dict` so that `AutoExtractAdapter` can include them in the resultant item. This ensures supporting new `AutoExtract` fields even if the item library is not in sync.

autoextract_poet.items.ArticleFromList

class `ArticleFromList(url: Optional[str] = None, probability: Optional[float] = None, headline: Optional[str] = None, datePublished: Optional[str] = None, datePublishedRaw: Optional[str] = None, author: Optional[str] = None, authorsList: List[str] = NOTHING, inLanguage: Optional[str] = None, mainImage: Optional[str] = None, images: List[str] = NOTHING, articleBody: Optional[str] = None)`

Bases: `autoextract_poet.items.Item`

__init__(url: *Optional*[str] = None, probability: *Optional*[float] = None, headline: *Optional*[str] = None, datePublished: *Optional*[str] = None, datePublishedRaw: *Optional*[str] = None, author: *Optional*[str] = None, authorsList: *List*[str] = NOTHING, inLanguage: *Optional*[str] = None, mainImage: *Optional*[str] = None, images: *List*[str] = NOTHING, articleBody: *Optional*[str] = None) → None

Method generated by attrs for class `ArticleFromList`.

Methods

<code>__init__</code> (url, probability, headline, ...)	Method generated by attrs for class <code>ArticleFromList</code> .
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

<code>url</code>
<code>probability</code>
<code>headline</code>
<code>datePublished</code>
<code>datePublishedRaw</code>
<code>author</code>
<code>authorsList</code>
<code>inLanguage</code>
<code>mainImage</code>

continues on next page

Table 16 – continued from previous page

images
articleBody

autoextract_poet.items.ArticleList

```
class ArticleList(url: Optional[str] = None, articles: List[autoextract_poet.items.ArticleFromList] =
    NOTHING, paginationNext: Optional[autoextract_poet.items.PaginationLink] = None,
    paginationPrevious: Optional[autoextract_poet.items.PaginationLink] = None)
    Bases: autoextract_poet.items.Item

    __init__(url: Optional[str] = None, articles: List[autoextract_poet.items.ArticleFromList] = NOTHING,
        paginationNext: Optional[autoextract_poet.items.PaginationLink] = None, paginationPrevious:
        Optional[autoextract_poet.items.PaginationLink] = None) → None
    Method generated by attrs for class ArticleList.
```

Methods

<code>__init__</code> (url, articles, paginationNext, ...)	Method generated by attrs for class ArticleList.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

url
articles
paginationNext
paginationPrevious

classmethod `from_dict`(item: Optional[Dict])

Read an item from a dictionary.

Unknown attributes are kept in the dict `_unknown_fields_dict` so that `AutoExtractAdapter` can include them in the resultant item. This ensures supporting new `AutoExtract` fields even if the item library is not in sync.

autoextract_poet.items.AvailableAtOrFrom

class AvailableAtOrFrom(raw: *Optional[str]* = None)
 Bases: *autoextract_poet.items.Item*
__init__(raw: *Optional[str]* = None) → None
 Method generated by attrs for class AvailableAtOrFrom.

Methods

<code>__init__</code> ([raw])	Method generated by attrs for class AvailableAtOrFrom.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

<code>raw</code>

autoextract_poet.items.Breadcrumb

class Breadcrumb(name: *Optional[str]* = None, link: *Optional[str]* = None)
 Bases: *autoextract_poet.items.Item*
__init__(name: *Optional[str]* = None, link: *Optional[str]* = None) → None
 Method generated by attrs for class Breadcrumb.

Methods

<code>__init__</code> ([name, link])	Method generated by attrs for class Breadcrumb.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

<code>name</code>
<code>link</code>

autoextract_poet.items.Comment

```
class Comment(probability: Optional[float] = None, text: Optional[str] = None, datePublished: Optional[str] =
    None, datePublishedRaw: Optional[str] = None, upvoteCount: Optional[int] = None,
    downvoteCount: Optional[int] = None)
```

Bases: `autoextract_poet.items.Item`

```
__init__(probability: Optional[float] = None, text: Optional[str] = None, datePublished: Optional[str] =
    None, datePublishedRaw: Optional[str] = None, upvoteCount: Optional[int] = None,
    downvoteCount: Optional[int] = None) → None
```

Method generated by attrs for class Comment.

Methods

<code>__init__([probability, text, datePublished, ...])</code>	Method generated by attrs for class Comment.
<code>from_dict(item)</code>	Read an item from a dictionary.
<code>from_list(items)</code>	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

<code>probability</code>
<code>text</code>
<code>datePublished</code>
<code>datePublishedRaw</code>
<code>upvoteCount</code>
<code>downvoteCount</code>

autoextract_poet.items.Comments

```
class Comments(url: Optional[str] = None, comments: List[autoextract_poet.items.Comment] = NOTHING)
    Bases: autoextract_poet.items.Item
```

```
__init__(url: Optional[str] = None, comments: List[autoextract_poet.items.Comment] = NOTHING) →
    None
```

Method generated by attrs for class Comments.

Methods

<code>__init__([url, comments])</code>	Method generated by attrs for class Comments.
<code>from_dict(item)</code>	Read an item from a dictionary.
<code>from_list(items)</code>	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

<code>url</code>
<code>comments</code>

classmethod `from_dict(item: Optional[Dict])`

Read an item from a dictionary.

Unknown attributes are kept in the dict `_unknown_fields_dict` so that `AutoExtractAdapter` can include them in the resultant item. This ensures supporting new `AutoExtract` fields even if the item library is not in sync.

autoextract_poet.items.ForumPost

class `ForumPost(probability: Optional[float] = None, text: Optional[str] = None, datePublished: Optional[str] = None, datePublishedRaw: Optional[str] = None, upvoteCount: Optional[int] = None, downvoteCount: Optional[int] = None, replyCount: Optional[int] = None)`

Bases: `autoextract_poet.items.Item`

`__init__(probability: Optional[float] = None, text: Optional[str] = None, datePublished: Optional[str] = None, datePublishedRaw: Optional[str] = None, upvoteCount: Optional[int] = None, downvoteCount: Optional[int] = None, replyCount: Optional[int] = None) → None`

Method generated by attrs for class ForumPost.

Methods

<code>__init__([probability, text, datePublished, ...])</code>	Method generated by attrs for class ForumPost.
<code>from_dict(item)</code>	Read an item from a dictionary.
<code>from_list(items)</code>	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

probability

text

datePublished

datePublishedRaw

upvoteCount

downvoteCount

replyCount

autoextract_poet.items.ForumPosts

class **ForumPosts**(*url: Optional[str] = None, topic: Optional[autoextract_poet.items.Topic] = None, posts: List[autoextract_poet.items.ForumPost] = NOTHING*)

Bases: *autoextract_poet.items.Item*

__init__(*url: Optional[str] = None, topic: Optional[autoextract_poet.items.Topic] = None, posts: List[autoextract_poet.items.ForumPost] = NOTHING*) → None

Method generated by attrs for class ForumPosts.

Methods

<code>__init__</code> ([url, topic, posts])	Method generated by attrs for class ForumPosts.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

url

topic

posts

classmethod **from_dict**(*item: Optional[Dict]*)

Read an item from a dictionary.

Unknown attributes are kept in the dict `_unknown_fields_dict` so that `AutoExtractAdapter` can include them in the resultant item. This ensures supporting new `AutoExtract` fields even if the item library is not in sync.

autoextract_poet.items.FuelEfficiency

```
class FuelEfficiency(raw: Optional[str] = None)
    Bases: autoextract_poet.items.Item
    __init__(raw: Optional[str] = None) → None
        Method generated by attrs for class FuelEfficiency.
```

Methods

<code>__init__([raw])</code>	Method generated by attrs for class FuelEfficiency.
<code>from_dict(item)</code>	Read an item from a dictionary.
<code>from_list(items)</code>	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

<code>raw</code>

autoextract_poet.items.GTIN

```
class GTIN(type: str, value: str)
    Bases: autoextract_poet.items.Item
    __init__(type: str, value: str) → None
        Method generated by attrs for class GTIN.
```

Methods

<code>__init__(type, value)</code>	Method generated by attrs for class GTIN.
<code>from_dict(item)</code>	Read an item from a dictionary.
<code>from_list(items)</code>	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

<code>type</code>
<code>value</code>

autoextract_poet.items.Item**class Item**Bases: `autoextract_poet.items._ItemBase``__init__()` → `None`

Method generated by attrs for class Item.

Methods

<code>__init__()</code>	Method generated by attrs for class Item.
<code>from_dict(item)</code>	Read an item from a dictionary.
<code>from_list(items)</code>	Read items from a list, invoking <code>from_dict</code> for each item in the list

classmethod `from_dict(item: Optional[Dict])`

Read an item from a dictionary.

Unknown attributes are kept in the dict `_unknown_fields_dict` so that `AutoExtractAdapter` can include them in the resultant item. This ensures supporting new `AutoExtract` fields even if the item library is not in sync.

classmethod `from_list(items: Optional[List[Dict]])` → `List`Read items from a list, invoking `from_dict` for each item in the list**autoextract_poet.items.JobPosting**

class JobPosting(*probability: *Optional[float]* = None, url: *Optional[str]* = None, title: *Optional[str]* = None, datePosted: *Optional[str]* = None, validThrough: *Optional[str]* = None, description: *Optional[str]* = None, descriptionHtml: *Optional[str]* = None, employmentType: *Optional[str]* = None, hiringOrganization: *Optional[autoextract_poet.items.Organization]* = None, baseSalary: *Optional[autoextract_poet.items.Salary]* = None, jobLocation: *Optional[autoextract_poet.items.Location]* = None)*

Bases: `autoextract_poet.items.Item`

`__init__`(*probability: *Optional[float]* = None, url: *Optional[str]* = None, title: *Optional[str]* = None, datePosted: *Optional[str]* = None, validThrough: *Optional[str]* = None, description: *Optional[str]* = None, descriptionHtml: *Optional[str]* = None, employmentType: *Optional[str]* = None, hiringOrganization: *Optional[autoextract_poet.items.Organization]* = None, baseSalary: *Optional[autoextract_poet.items.Salary]* = None, jobLocation: *Optional[autoextract_poet.items.Location]* = None*) → `None`

Method generated by attrs for class JobPosting.

Methods

<code>__init__</code> ([probability, url, title, ...])	Method generated by attrs for class JobPosting.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

probability
url
title
datePosted
validThrough
description
descriptionHtml
employmentType
hiringOrganization
baseSalary
jobLocation

classmethod `from_dict`(item: *Optional[Dict]*)

Read an item from a dictionary.

Unknown attributes are kept in the dict `_unknown_fields_dict` so that `AutoExtractAdapter` can include them in the resultant item. This ensures supporting new `AutoExtract` fields even if the item library is not in sync.

`autoextract_poet.items.Location`

class `Location`(raw: *Optional[str]* = None)

Bases: `autoextract_poet.items.Item`

`__init__`(raw: *Optional[str]* = None) → None

Method generated by attrs for class Location.

Methods

<code>__init__([raw])</code>	Method generated by attrs for class Location.
<code>from_dict(item)</code>	Read an item from a dictionary.
<code>from_list(items)</code>	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

<code>raw</code>

autoextract_poet.items.MileageFromOdometer

class `MileageFromOdometer`(*value: Optional[int] = None, unitCode: Optional[str] = None*)

Bases: `autoextract_poet.items.Item`

`__init__`(*value: Optional[int] = None, unitCode: Optional[str] = None*) → `None`

Method generated by attrs for class MileageFromOdometer.

Methods

<code>__init__([value, unitCode])</code>	Method generated by attrs for class MileageFromOdometer.
<code>from_dict(item)</code>	Read an item from a dictionary.
<code>from_list(items)</code>	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

<code>value</code>
<code>unitCode</code>

autoextract_poet.items.Offer

class `Offer`(*price: Optional[str] = None, currency: Optional[str] = None, availability: Optional[str] = None, regularPrice: Optional[str] = None*)

Bases: `autoextract_poet.items.Item`

`__init__`(*price: Optional[str] = None, currency: Optional[str] = None, availability: Optional[str] = None, regularPrice: Optional[str] = None*) → `None`

Method generated by attrs for class Offer.

Methods

<code>__init__([price, currency, availability, ...])</code>	Method generated by attrs for class Offer.
<code>from_dict(item)</code>	Read an item from a dictionary.
<code>from_list(items)</code>	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

<code>price</code>
<code>currency</code>
<code>availability</code>
<code>regularPrice</code>

autoextract_poet.items.Organization

class Organization(*raw: Optional[str] = None*)

Bases: `autoextract_poet.items.Item`

`__init__(raw: Optional[str] = None) → None`

Method generated by attrs for class Organization.

Methods

<code>__init__([raw])</code>	Method generated by attrs for class Organization.
<code>from_dict(item)</code>	Read an item from a dictionary.
<code>from_list(items)</code>	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

<code>raw</code>

autoextract_poet.items.PaginationLink

class **PaginationLink**(url: *Optional[str] = None*, text: *Optional[str] = None*)

Bases: `autoextract_poet.items.Item`

__init__(url: *Optional[str] = None*, text: *Optional[str] = None*) → *None*

Method generated by attrs for class `PaginationLink`.

Methods

<code>__init__</code> ([url, text])	Method generated by attrs for class <code>PaginationLink</code> .
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

url
text

autoextract_poet.items.Product

class **Product**(url: *Optional[str] = None*, canonicalUrl: *Optional[str] = None*, probability: *Optional[float] = None*, name: *Optional[str] = None*, offers: *List[autoextract_poet.items.Offer] = NOTHING*, sku: *Optional[str] = None*, gtin: *List[autoextract_poet.items.GTIN] = NOTHING*, mpn: *Optional[str] = None*, brand: *Optional[str] = None*, breadcrumbs: *List[autoextract_poet.items.Breadcrumb] = NOTHING*, mainImage: *Optional[str] = None*, images: *List[str] = NOTHING*, description: *Optional[str] = None*, descriptionHtml: *Optional[str] = None*, additionalProperty: *List[autoextract_poet.items.AdditionalProperty] = NOTHING*, aggregateRating: *Optional[autoextract_poet.items.Rating] = None*, color: *Optional[str] = None*, size: *Optional[str] = None*, style: *Optional[str] = None*)

Bases: `autoextract_poet.items.Item`

__init__(url: *Optional[str] = None*, canonicalUrl: *Optional[str] = None*, probability: *Optional[float] = None*, name: *Optional[str] = None*, offers: *List[autoextract_poet.items.Offer] = NOTHING*, sku: *Optional[str] = None*, gtin: *List[autoextract_poet.items.GTIN] = NOTHING*, mpn: *Optional[str] = None*, brand: *Optional[str] = None*, breadcrumbs: *List[autoextract_poet.items.Breadcrumb] = NOTHING*, mainImage: *Optional[str] = None*, images: *List[str] = NOTHING*, description: *Optional[str] = None*, descriptionHtml: *Optional[str] = None*, additionalProperty: *List[autoextract_poet.items.AdditionalProperty] = NOTHING*, aggregateRating: *Optional[autoextract_poet.items.Rating] = None*, color: *Optional[str] = None*, size: *Optional[str] = None*, style: *Optional[str] = None*) → *None*

Method generated by attrs for class `Product`.

Methods

<code>__init__</code> (url, canonicalUrl, probability, ...)	Method generated by attrs for class Product.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

url
canonicalUrl
probability
name
offers
sku
gtin
mpn
brand
breadcrumbs
mainImage
images
description
descriptionHtml
additionalProperty
aggregateRating
color
size
style

classmethod `from_dict`(item: *Optional[Dict]*)
Read an item from a dictionary.

Unknown attributes are kept in the dict `_unknown_fields_dict` so that `AutoExtractAdapter` can include them in the resultant item. This ensures supporting new `AutoExtract` fields even if the item library is not in sync.

autoextract_poet.items.ProductFromList

```
class ProductFromList(probability: Optional[float] = None, url: Optional[str] = None, name: Optional[str] =
    None, offers: List[autoextract_poet.items.Offer] = NOTHING, sku: Optional[str] =
    None, brand: Optional[str] = None, mainImage: Optional[str] = None, images:
    List[str] = NOTHING, description: Optional[str] = None, aggregateRating:
    Optional[autoextract_poet.items.Rating] = None)
```

Bases: `autoextract_poet.items.Item`

```
__init__(probability: Optional[float] = None, url: Optional[str] = None, name: Optional[str] = None,
    offers: List[autoextract_poet.items.Offer] = NOTHING, sku: Optional[str] = None, brand:
    Optional[str] = None, mainImage: Optional[str] = None, images: List[str] = NOTHING,
    description: Optional[str] = None, aggregateRating: Optional[autoextract_poet.items.Rating] =
    None) → None
```

Method generated by attrs for class `ProductFromList`.

Methods

<code>__init__([probability, url, name, offers, ...])</code>	Method generated by attrs for class <code>ProductFromList</code> .
<code>from_dict(item)</code>	Read an item from a dictionary.
<code>from_list(items)</code>	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

<code>probability</code>
<code>url</code>
<code>name</code>
<code>offers</code>
<code>sku</code>
<code>brand</code>
<code>mainImage</code>
<code>images</code>
<code>description</code>
<code>aggregateRating</code>

classmethod `from_dict(item: Optional[Dict])`

Read an item from a dictionary.

Unknown attributes are kept in the dict `_unknown_fields_dict` so that `AutoExtractAdapter` can include them in the resultant item. This ensures supporting new `AutoExtract` fields even if the item library is not in sync.

autoextract_poet.items.ProductList

class `ProductList(url: Optional[str] = None, products: List[autoextract_poet.items.ProductFromList] = NOTHING, breadcrumbs: List[autoextract_poet.items.Breadcrumb] = NOTHING, paginationNext: Optional[autoextract_poet.items.PaginationLink] = None, paginationPrevious: Optional[autoextract_poet.items.PaginationLink] = None)`

Bases: `autoextract_poet.items.Item`

__init__(url: *Optional*[str] = None, products: List[autoextract_poet.items.ProductFromList] = NOTHING, breadcrumbs: List[autoextract_poet.items.Breadcrumb] = NOTHING, paginationNext: *Optional*[autoextract_poet.items.PaginationLink] = None, paginationPrevious: *Optional*[autoextract_poet.items.PaginationLink] = None) → None

Method generated by attrs for class `ProductList`.

Methods

<code>__init__</code> ([url, products, breadcrumbs, ...])	Method generated by attrs for class <code>ProductList</code> .
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

<code>url</code>
<code>products</code>
<code>breadcrumbs</code>
<code>paginationNext</code>
<code>paginationPrevious</code>

classmethod `from_dict(item: Optional[Dict])`

Read an item from a dictionary.

Unknown attributes are kept in the dict `_unknown_fields_dict` so that `AutoExtractAdapter` can include them in the resultant item. This ensures supporting new `AutoExtract` fields even if the item library is not in sync.

autoextract_poet.items.Rating

```
class Rating(ratingValue: Optional[float] = None, bestRating: Optional[float] = None, reviewCount:
    Optional[int] = None)
    Bases: autoextract_poet.items.Item
    __init__(ratingValue: Optional[float] = None, bestRating: Optional[float] = None, reviewCount:
        Optional[int] = None) → None
        Method generated by attrs for class Rating.
```

Methods

<code>__init__</code> ([ratingValue, bestRating, reviewCount])	Method generated by attrs for class Rating.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

<code>ratingValue</code>
<code>bestRating</code>
<code>reviewCount</code>

autoextract_poet.items.RealEstate

```
class RealEstate(url: Optional[str] = None, probability: Optional[float] = None, name: Optional[str] = None,
    datePublished: Optional[str] = None, datePublishedRaw: Optional[str] = None, description:
    Optional[str] = None, mainImage: Optional[str] = None, images: List[str] = NOTHING,
    yearBuilt: Optional[int] = None, breadcrumbs: List[autoextract_poet.items.Breadcrumb] =
    NOTHING, additionalProperty: List[autoextract_poet.items.AdditionalProperty] =
    NOTHING, address: Optional[autoextract_poet.items.Address] = None, area:
    Optional[autoextract_poet.items.Area] = None, numberOfBathroomsTotal: Optional[int] =
    None, numberOfFullBathrooms: Optional[int] = None, numberOfPartialBathrooms:
    Optional[int] = None, numberOfBedrooms: Optional[int] = None, numberOfRooms:
    Optional[int] = None, identifier: Optional[str] = None, tradeActions:
    List[autoextract_poet.items.TradeAction] = NOTHING)
    Bases: autoextract_poet.items.Item
```

```
__init__(url: Optional[str] = None, probability: Optional[float] = None, name: Optional[str] = None,
         datePublished: Optional[str] = None, datePublishedRaw: Optional[str] = None, description:
         Optional[str] = None, mainImage: Optional[str] = None, images: List[str] = NOTHING,
         yearBuilt: Optional[int] = None, breadcrumbs: List[autoextract_poet.items.Breadcrumb] =
         NOTHING, additionalProperty: List[autoextract_poet.items.AdditionalProperty] = NOTHING,
         address: Optional[autoextract_poet.items.Address] = None, area:
         Optional[autoextract_poet.items.Area] = None, numberOfBathroomsTotal: Optional[int] = None,
         numberOfFullBathrooms: Optional[int] = None, numberOfPartialBathrooms: Optional[int] =
         None, numberOfBedrooms: Optional[int] = None, numberOfRooms: Optional[int] = None,
         identifier: Optional[str] = None, tradeActions: List[autoextract_poet.items.TradeAction] =
         NOTHING) → None
```

Method generated by attrs for class RealEstate.

Methods

<code>__init__</code> (url, probability, name, ...)	Method generated by attrs for class RealEstate.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

<code>url</code>
<code>probability</code>
<code>name</code>
<code>datePublished</code>
<code>datePublishedRaw</code>
<code>description</code>
<code>mainImage</code>
<code>images</code>
<code>yearBuilt</code>
<code>breadcrumbs</code>
<code>additionalProperty</code>
<code>address</code>
<code>area</code>

continues on next page

Table 57 – continued from previous page

numberOfBathroomsTotal
numberOfFullBathrooms
numberOfPartialBathrooms
numberOfBedrooms
numberOfRooms
identifier
tradeActions

classmethod `from_dict`(item: *Optional[Dict]*)

Read an item from a dictionary.

Unknown attributes are kept in the dict `_unknown_fields_dict` so that `AutoExtractAdapter` can include them in the resultant item. This ensures supporting new `AutoExtract` fields even if the item library is not in sync.

autoextract_poet.items.Review

class `Review`(name: *Optional[str]* = None, reviewBody: *Optional[str]* = None, reviewRating: *Optional[autoextract_poet.items.Rating]* = None, datePublished: *Optional[str]* = None, datePublishedRaw: *Optional[str]* = None, votedHelpful: *Optional[int]* = None, votedUnhelpful: *Optional[int]* = None, isVerified: *Optional[bool]* = None, probability: *Optional[float]* = None)
 Bases: `autoextract_poet.items.Item`

__init__(name: *Optional[str]* = None, reviewBody: *Optional[str]* = None, reviewRating: *Optional[autoextract_poet.items.Rating]* = None, datePublished: *Optional[str]* = None, datePublishedRaw: *Optional[str]* = None, votedHelpful: *Optional[int]* = None, votedUnhelpful: *Optional[int]* = None, isVerified: *Optional[bool]* = None, probability: *Optional[float]* = None) → None

Method generated by attrs for class Review.

Methods

<code>__init__</code> ([name, reviewBody, reviewRating, ...])	Method generated by attrs for class Review.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

name
reviewBody
reviewRating
datePublished
datePublishedRaw
votedHelpful
votedUnhelpful
isVerified
probability

classmethod `from_dict(item: Optional[Dict])`

Read an item from a dictionary.

Unknown attributes are kept in the dict `_unknown_fields_dict` so that `AutoExtractAdapter` can include them in the resultant item. This ensures supporting new `AutoExtract` fields even if the item library is not in sync.

autoextract_poet.items.Reviews

class `Reviews(url: Optional[str] = None, reviews: List[autoextract_poet.items.Review] = NOTHING, paginationNext: Optional[autoextract_poet.items.PaginationLink] = None, paginationPrevious: Optional[autoextract_poet.items.PaginationLink] = None)`

Bases: `autoextract_poet.items.Item`

__init__(url: *Optional[str]* = None, reviews: List[autoextract_poet.items.Review] = NOTHING, paginationNext: *Optional[autoextract_poet.items.PaginationLink]* = None, paginationPrevious: *Optional[autoextract_poet.items.PaginationLink]* = None) → None

Method generated by attrs for class Reviews.

Methods

<code>__init__</code> (url, reviews, paginationNext, ...)	Method generated by attrs for class Reviews.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

url

reviews

paginationNext

paginationPrevious

classmethod `from_dict(item: Optional[Dict])`

Read an item from a dictionary.

Unknown attributes are kept in the dict `_unknown_fields_dict` so that `AutoExtractAdapter` can include them in the resultant item. This ensures supporting new `AutoExtract` fields even if the item library is not in sync.

autoextract_poet.items.Salary

class `Salary(raw: Optional[str] = None, value: Optional[float] = None, currency: Optional[str] = None)`

Bases: `autoextract_poet.items.Item`

__init__(raw: *Optional[str]* = None, value: *Optional[float]* = None, currency: *Optional[str]* = None) → None

Method generated by attrs for class Salary.

Methods

<code>__init__</code> ([raw, value, currency])	Method generated by attrs for class Salary.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

raw

value

currency

autoextract_poet.items.Topic

class Topic(name: *Optional[str]* = None)
 Bases: `autoextract_poet.items.Item`
__init__(name: *Optional[str]* = None) → None
 Method generated by attrs for class Topic.

Methods

<code>__init__</code> ([name])	Method generated by attrs for class Topic.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

name

autoextract_poet.items.TradeAction

class TradeAction(tradeType: *Optional[str]* = None, price: *Optional[str]* = None, currency: *Optional[str]* = None)
 Bases: `autoextract_poet.items.Item`
__init__(tradeType: *Optional[str]* = None, price: *Optional[str]* = None, currency: *Optional[str]* = None) → None
 Method generated by attrs for class TradeAction.

Methods

<code>__init__</code> ([tradeType, price, currency])	Method generated by attrs for class TradeAction.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

tradeType
price
currency

autoextract_poet.items.Vehicle

```
class Vehicle(url: Optional[str] = None, canonicalUrl: Optional[str] = None, probability: Optional[float] =
None, name: Optional[str] = None, offers: List[autoextract_poet.items.Offer] = NOTHING, sku:
Optional[str] = None, mpn: Optional[str] = None, brand: Optional[str] = None, breadcrumbs:
List[autoextract_poet.items.Breadcrumb] = NOTHING, mainImage: Optional[str] = None,
images: List[str] = NOTHING, description: Optional[str] = None, descriptionHtml: Optional[str]
= None, additionalProperty: List[autoextract_poet.items.AdditionalProperty] = NOTHING,
aggregateRating: Optional[autoextract_poet.items.Rating] = None, vehicleIdentificationNumber:
Optional[str] = None, mileageFromOdometer:
Optional[autoextract_poet.items.MileageFromOdometer] = None, vehicleTransmission:
Optional[str] = None, fuelType: Optional[str] = None, vehicleEngine:
Optional[autoextract_poet.items.VehicleEngine] = None, color: Optional[str] = None,
vehicleInteriorColor: Optional[str] = None, availableAtOrFrom:
Optional[autoextract_poet.items.AvailableAtOrFrom] = None, numberOfDoors: Optional[int] =
None, vehicleSeatingCapacity: Optional[int] = None, fuelEfficiency:
List[autoextract_poet.items.FuelEfficiency] = NOTHING)
```

Bases: `autoextract_poet.items.Item`

```
__init__(url: Optional[str] = None, canonicalUrl: Optional[str] = None, probability: Optional[float] =
None, name: Optional[str] = None, offers: List[autoextract_poet.items.Offer] = NOTHING, sku:
Optional[str] = None, mpn: Optional[str] = None, brand: Optional[str] = None, breadcrumbs:
List[autoextract_poet.items.Breadcrumb] = NOTHING, mainImage: Optional[str] = None,
images: List[str] = NOTHING, description: Optional[str] = None, descriptionHtml: Optional[str]
= None, additionalProperty: List[autoextract_poet.items.AdditionalProperty] = NOTHING,
aggregateRating: Optional[autoextract_poet.items.Rating] = None, vehicleIdentificationNumber:
Optional[str] = None, mileageFromOdometer:
Optional[autoextract_poet.items.MileageFromOdometer] = None, vehicleTransmission:
Optional[str] = None, fuelType: Optional[str] = None, vehicleEngine:
Optional[autoextract_poet.items.VehicleEngine] = None, color: Optional[str] = None,
vehicleInteriorColor: Optional[str] = None, availableAtOrFrom:
Optional[autoextract_poet.items.AvailableAtOrFrom] = None, numberOfDoors: Optional[int] =
None, vehicleSeatingCapacity: Optional[int] = None, fuelEfficiency:
List[autoextract_poet.items.FuelEfficiency] = NOTHING) → None
```

Method generated by attrs for class Vehicle.

Methods

<code>__init__</code> (url, canonicalUrl, probability, ...)	Method generated by attrs for class Vehicle.
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

url

canonicalUrl

probability

name

offers

sku

mpn

brand

breadcrumbs

mainImage

images

description

descriptionHtml

additionalProperty

aggregateRating

vehicleIdentificationNumber

mileageFromOdometer

vehicleTransmission

fuelType

vehicleEngine

color

vehicleInteriorColor

availableAtOrFrom

numberOfDoors

continues on next page

Table 69 – continued from previous page

vehicleSeatingCapacity

fuelEfficiency

classmethod `from_dict(item: Optional[Dict])`

Read an item from a dictionary.

Unknown attributes are kept in the dict `_unknown_fields_dict` so that `AutoExtractAdapter` can include them in the resultant item. This ensures supporting new `AutoExtract` fields even if the item library is not in sync.

autoextract_poet.items.VehicleEngine

class `VehicleEngine(raw: Optional[str] = None)`

Bases: `autoextract_poet.items.Item`
__init__(raw: *Optional[str]* = None) → None

Method generated by attrs for class `VehicleEngine`.

Methods

<code>__init__</code> ([raw])	Method generated by attrs for class <code>VehicleEngine</code> .
<code>from_dict</code> (item)	Read an item from a dictionary.
<code>from_list</code> (items)	Read items from a list, invoking <code>from_dict</code> for each item in the list

Attributes

raw

2.1.3 autoextract_poet.page_inputs

Functions

<code>get_item_class</code> (page_input_cls)	Return item class for the page input class.
--	---

autoextract_poet.page_inputs.get_item_class

get_item_class(*page_input_cls*: *Type*[autoextract_poet.page_inputs.AutoExtractData]) → *Type*[autoextract_poet.items.Item]

Return item class for the page input class.

```
>>> get_item_class(AutoExtractArticleData) is Article
True
>>> get_item_class(AutoExtractProductData) is Product
True
>>> get_item_class(AutoExtractData) is T
True
```

Classes

<i>AutoExtractArticleData</i> (data)	Container for AutoExtract Article data.
<i>AutoExtractArticleListData</i> (data)	Container for AutoExtract Article list data.
<i>AutoExtractCommentsData</i> (data)	Container for AutoExtract Comments data.
<i>AutoExtractData</i> (data)	Container for AutoExtract data.
<i>AutoExtractForumPostsData</i> (data)	Container for AutoExtract Forum Posts data.
<i>AutoExtractHtml</i> (url, html)	A container for URL and HTML content retrieved from AutoExtract.
<i>AutoExtractJobPostingData</i> (data)	Container for AutoExtract Job Posting data.
<i>AutoExtractProductData</i> (data)	Container for AutoExtract Product data.
<i>AutoExtractProductListData</i> (data)	Container for AutoExtract Product list data.
<i>AutoExtractRealEstateData</i> (data)	Container for AutoExtract Real Estate data.
<i>AutoExtractReviewsData</i> (data)	Container for AutoExtract Reviews data.
<i>AutoExtractVehicleData</i> (data)	Container for AutoExtract Vehicle data.

autoextract_poet.page_inputs.AutoExtractArticleData

class *AutoExtractArticleData*(*data*: *dict*)

Bases: *autoextract_poet.page_inputs.AutoExtractData*[*autoextract_poet.items.Article*]

Container for AutoExtract Article data.

<https://docs.zyte.com/automatic-extraction/article.html>

__init__(*data*: *dict*) → *None*

Method generated by attrs for class AutoExtractArticleData.

Methods

__init__ (data)	Method generated by attrs for class AutoExtractArticleData.
to_item ()	

Attributes

item_class

page_type

autoextract_poet.page_inputs.AutoExtractArticleListData

class AutoExtractArticleListData(*data: dict*)

Bases: `autoextract_poet.page_inputs.AutoExtractData[autoextract_poet.items.ArticleList]`

Container for AutoExtract Article list data.

<https://docs.zyte.com/automatic-extraction/article-list.html>

__init__(*data: dict*) → None

Method generated by attrs for class AutoExtractArticleListData.

Methods

__init__(*data*)

Method generated by attrs for class AutoExtractArticleListData.

to_item()

Attributes

item_class

page_type

autoextract_poet.page_inputs.AutoExtractCommentsData

class AutoExtractCommentsData(*data: dict*)

Bases: `autoextract_poet.page_inputs.AutoExtractData[autoextract_poet.items.Comments]`

Container for AutoExtract Comments data.

<https://docs.zyte.com/automatic-extraction/comment.html>

__init__(*data: dict*) → None

Method generated by attrs for class AutoExtractCommentsData.

Methods

<code>__init__(data)</code>	Method generated by attrs for class AutoExtractCommentsData.
<code>to_item()</code>	

Attributes

<code>item_class</code>
<code>page_type</code>

autoextract_poet.page_inputs.AutoExtractData

class AutoExtractData(*data: dict*)

Bases: `Generic[autoextract_poet.page_inputs.T]`

Container for AutoExtract data.

Should not be used directly by providers. Use derived classes like AutoExtractArticleData and similar.

API responses are wrapped in a JSON array (this is to facilitate query batching) but we're receiving single responses here..

<https://docs.zyte.com/automatic-extraction.html#responses>

`__init__(data: dict) → None`

Method generated by attrs for class AutoExtractData.

Methods

<code>__init__(data)</code>	Method generated by attrs for class AutoExtractData.
<code>to_item()</code>	

Attributes

<code>item_class</code>
<code>page_type</code>
<code>data</code>

autoextract_poet.page_inputs.AutoExtractForumPostsData**class AutoExtractForumPostsData**(*data: dict*)Bases: `autoextract_poet.page_inputs.AutoExtractData[autoextract_poet.items.ForumPosts]`

Container for AutoExtract Forum Posts data.

<https://docs.zyte.com/automatic-extraction/forum-post.html>**__init__**(*data: dict*) → `None`

Method generated by attrs for class AutoExtractForumPostsData.

Methods

__init__ (<i>data</i>)	Method generated by attrs for class AutoExtractForumPostsData.
to_item ()	

Attributes

item_class
page_type

autoextract_poet.page_inputs.AutoExtractHtml**class AutoExtractHtml**(*url: str, html: str*)Bases: `object`

A container for URL and HTML content retrieved from AutoExtract.

url should be an URL of the response (after all redirects), not an URL of the request, if possible.*html* should be browser HTML in unicode**__init__**(*url: str, html: str*) → `None`

Method generated by attrs for class AutoExtractHtml.

Methods

__init__ (<i>url, html</i>)	Method generated by attrs for class AutoExtractHtml.
--------------------------------------	--

Attributes

url

html

autoextract_poet.page_inputs.AutoExtractJobPostingData

class AutoExtractJobPostingData(*data: dict*)

Bases: `autoextract_poet.page_inputs.AutoExtractData[autoextract_poet.items.JobPosting]`

Container for AutoExtract Job Posting data.

<https://docs.zyte.com/automatic-extraction/job-posting.html>

__init__(*data: dict*) → None

Method generated by attrs for class AutoExtractJobPostingData.

Methods

__init__(*data*)

Method generated by attrs for class AutoExtractJobPostingData.

to_item()

Attributes

item_class

page_type

autoextract_poet.page_inputs.AutoExtractProductData

class AutoExtractProductData(*data: dict*)

Bases: `autoextract_poet.page_inputs.AutoExtractData[autoextract_poet.items.Product]`

Container for AutoExtract Product data.

<https://docs.zyte.com/automatic-extraction/product.html>

__init__(*data: dict*) → None

Method generated by attrs for class AutoExtractProductData.

Methods

<code>__init__(data)</code>	Method generated by attrs for class AutoExtractProductData.
<code>to_item()</code>	

Attributes

<code>item_class</code>	
<code>page_type</code>	

autoextract_poet.page_inputs.AutoExtractProductListData

```
class AutoExtractProductListData(data: dict)
    Bases: autoextract_poet.page_inputs.AutoExtractData[autoextract_poet.items.
    ProductList]
    Container for AutoExtract Product list data.
    https://docs.zyte.com/automatic-extraction/product-list.html
    __init__(data: dict) → None
        Method generated by attrs for class AutoExtractProductListData.
```

Methods

<code>__init__(data)</code>	Method generated by attrs for class AutoExtractProductListData.
<code>to_item()</code>	

Attributes

<code>item_class</code>	
<code>page_type</code>	

autoextract_poet.page_inputs.AutoExtractRealEstateData

class AutoExtractRealEstateData(*data: dict*)

Bases: `autoextract_poet.page_inputs.AutoExtractData[autoextract_poet.items.RealEstate]`

Container for AutoExtract Real Estate data.

<https://docs.zyte.com/automatic-extraction/real-estate.html>

__init__(*data: dict*) → None

Method generated by attrs for class AutoExtractRealEstateData.

Methods

<code>__init__</code> (data)	Method generated by attrs for class AutoExtractRealEstateData.
<code>to_item</code> ()	

Attributes

<code>item_class</code>	
<code>page_type</code>	

autoextract_poet.page_inputs.AutoExtractReviewsData

class AutoExtractReviewsData(*data: dict*)

Bases: `autoextract_poet.page_inputs.AutoExtractData[autoextract_poet.items.Reviews]`

Container for AutoExtract Reviews data.

<https://docs.zyte.com/automatic-extraction/review.html>

__init__(*data: dict*) → None

Method generated by attrs for class AutoExtractReviewsData.

Methods

<code>__init__</code> (data)	Method generated by attrs for class AutoExtractReviewsData.
<code>to_item</code> ()	

Attributes

`item_class`

`page_type`

autoextract_poet.page_inputs.AutoExtractVehicleData

class `AutoExtractVehicleData`(*data: dict*)

Bases: `autoextract_poet.page_inputs.AutoExtractData[autoextract_poet.items.Vehicle]`

Container for AutoExtract Vehicle data.

<https://docs.zyte.com/automatic-extraction/vehicle.html>

__init__(*data: dict*) → *None*

Method generated by attrs for class `AutoExtractVehicleData`.

Methods

`__init__`(*data*)

Method generated by attrs for class `AutoExtractVehicleData`.

`to_item`()

Attributes

`item_class`

`page_type`

2.1.4 autoextract_poet.pages

Classes

<code>AutoExtractArticleListPage</code> (<i>article_list_data</i>)	Article list data from AutoExtract
<code>AutoExtractArticlePage</code> (<i>article_data</i>)	Article data from AutoExtract
<code>AutoExtractCommentsPage</code> (<i>comments_data</i>)	Comments data from AutoExtract
<code>AutoExtractForumPostsPage</code> (<i>forum_posts_data</i>)	Forum posts data from AutoExtract
<code>AutoExtractItemWebPage</code> (<i>response</i>)	<code>AutoExtractWebPage</code> that requires the <code>to_item()</code> method to be implemented.
<code>AutoExtractJobPostingPage</code> (<i>job_posting_data</i>)	Job posting data from AutoExtract
<code>AutoExtractProductListPage</code> (<i>product_list_data</i>)	Product list data from AutoExtract
<code>AutoExtractProductPage</code> (<i>product_data</i>)	Product data from AutoExtract
<code>AutoExtractRealEstatePage</code> (<i>real_estate_data</i>)	Real estate data from AutoExtract

continues on next page

Table 98 – continued from previous page

<i>AutoExtractReviewsPage</i> (reviews_data)	Reviews data from AutoExtract
<i>AutoExtractVehiclePage</i> (vehicle_data)	Vehicle data from AutoExtract
<i>AutoExtractWebPage</i> (response)	Base Page Object which requires <i>AutoExtractHtml</i> and provides XPath / CSS shortcuts.

autoextract_poet.pages.AutoExtractArticleListPage

```
class AutoExtractArticleListPage(article_list_data:
                                autoextract_poet.page_inputs.AutoExtractArticleListData)
    Bases: web_poet.pages.ItemPage
    Article list data from AutoExtract
    https://docs.zyte.com/automatic-extraction/article-list.html
    __init__(article_list_data: autoextract_poet.page_inputs.AutoExtractArticleListData) → None
        Method generated by attrs for class AutoExtractArticleListPage.
```

Methods

<i>__init__</i> (article_list_data)	Method generated by attrs for class AutoExtractArticleListPage.
<i>to_item</i> ()	Extract an item from a web page

Attributes

article_list_data

[*to_item*](#)() → [*Optional*](#)[[*autoextract_poet.items.ArticleList*](#)]
 Extract an item from a web page

autoextract_poet.pages.AutoExtractArticlePage

```
class AutoExtractArticlePage(article_data: autoextract_poet.page_inputs.AutoExtractArticleData)
    Bases: web_poet.pages.ItemPage
    Article data from AutoExtract
    https://docs.zyte.com/automatic-extraction/article.html
    __init__(article_data: autoextract_poet.page_inputs.AutoExtractArticleData) → None
        Method generated by attrs for class AutoExtractArticlePage.
```


Methods

<code>__init__(article_data)</code>	Method generated by attrs for class AutoExtractArticlePage.
<code>to_item()</code>	Extract an item from a web page

Attributes

<code>article_data</code>

`to_item()` → `Optional[autoextract_poet.items.Article]`
Extract an item from a web page

autoextract_poet.pages.AutoExtractCommentsPage

class AutoExtractCommentsPage(*comments_data*: autoextract_poet.page_inputs.AutoExtractCommentsData)

Bases: `web_poet.pages.ItemPage`

Comments data from AutoExtract

<https://docs.zyte.com/automatic-extraction/comment.html>

`__init__(comments_data: autoextract_poet.page_inputs.AutoExtractCommentsData)` → `None`
Method generated by attrs for class AutoExtractCommentsPage.

Methods

<code>__init__(comments_data)</code>	Method generated by attrs for class AutoExtractCommentsPage.
<code>to_item()</code>	Extract an item from a web page

Attributes

<code>comments_data</code>

`to_item()` → `Optional[autoextract_poet.items.Comments]`
Extract an item from a web page

autoextract_poet.pages.AutoExtractForumPostsPage

```
class AutoExtractForumPostsPage(forum_posts_data:
                                autoextract_poet.page_inputs.AutoExtractForumPostsData)
    Bases: web_poet.pages.ItemPage
    Forum posts data from AutoExtract
    https://docs.zyte.com/automatic-extraction/forum-post.html
    __init__(forum_posts_data: autoextract_poet.page_inputs.AutoExtractForumPostsData) → None
        Method generated by attrs for class AutoExtractForumPostsPage.
```

Methods

<code>__init__(forum_posts_data)</code>	Method generated by attrs for class AutoExtractForumPostsPage.
<code>to_item()</code>	Extract an item from a web page

Attributes

<code>forum_posts_data</code>

`to_item()` → `Optional[autoextract_poet.items.ForumPosts]`
 Extract an item from a web page

autoextract_poet.pages.AutoExtractItemWebPage

```
class AutoExtractItemWebPage(response: autoextract_poet.page_inputs.AutoExtractHtml)
    Bases: autoextract_poet.pages.AutoExtractWebPage, web_poet.pages.ItemPage
    AutoExtractWebPage that requires the to_item() method to be implemented.
    __init__(response: autoextract_poet.page_inputs.AutoExtractHtml) → None
        Method generated by attrs for class AutoExtractItemWebPage.
```

Methods

<code>__init__(response)</code>	Method generated by attrs for class AutoExtractItemWebPage.
<code>css(query)</code>	Run a CSS query on a response, using <code>parsel.Selector</code> .
<code>to_item()</code>	Extract an item from a web page
<code>urljoin(url)</code>	Convert url to absolute, taking in account url and baseurl of the response
<code>xpath(query, **kwargs)</code>	Run an XPath query on a response, using <code>parsel.Selector</code> .

Attributes

<code>base_url</code>	Return the base url of the given response
<code>html</code>	Shortcut to HTML Response's content.
<code>selector</code>	<code>parsel.Selector</code> instance for the HTML Response.
<code>url</code>	Shortcut to HTML Response's URL.

autoextract_poet.pages.AutoExtractJobPostingPage

```
class AutoExtractJobPostingPage(job_posting_data:
                                autoextract_poet.page_inputs.AutoExtractJobPostingData)
    Bases: web_poet.pages.ItemPage
    Job posting data from AutoExtract
    https://docs.zyte.com/automatic-extraction/job-posting.html
    __init__(job_posting_data: autoextract_poet.page_inputs.AutoExtractJobPostingData) → None
        Method generated by attrs for class AutoExtractJobPostingPage.
```

Methods

<code>__init__(<i>job_posting_data</i>)</code>	Method generated by attrs for class AutoExtractJobPostingPage.
<code>to_item()</code>	Extract an item from a web page

Attributes

<code>job_posting_data</code>	
-------------------------------	--

```
to_item() → Optional[autoextract_poet.items.JobPosting]
    Extract an item from a web page
```

autoextract_poet.pages.AutoExtractProductListPage

```
class AutoExtractProductListPage(product_list_data:
                                   autoextract_poet.page_inputs.AutoExtractProductListData)
    Bases: web_poet.pages.ItemPage
    Product list data from AutoExtract
    https://docs.zyte.com/automatic-extraction/product-list.html
    __init__(product_list_data: autoextract_poet.page_inputs.AutoExtractProductListData) → None
        Method generated by attrs for class AutoExtractProductListPage.
```

Methods

<code>__init__(product_list_data)</code>	Method generated by attrs for class <code>AutoExtractProductListPage</code> .
<code>to_item()</code>	Extract an item from a web page

Attributes

<code>product_list_data</code>

`to_item()` → `Optional[autoextract_poet.items.ProductList]`
Extract an item from a web page

autoextract_poet.pages.AutoExtractProductPage

class `AutoExtractProductPage`(*product_data*: `autoextract_poet.page_inputs.AutoExtractProductData`)

Bases: `web_poet.pages.ItemPage`

Product data from AutoExtract

<https://docs.zyte.com/automatic-extraction/product.html>

`__init__(product_data: autoextract_poet.page_inputs.AutoExtractProductData)` → `None`
Method generated by attrs for class `AutoExtractProductPage`.

Methods

<code>__init__(product_data)</code>	Method generated by attrs for class <code>AutoExtractProductPage</code> .
<code>to_item()</code>	Extract an item from a web page

Attributes

<code>product_data</code>

`to_item()` → `Optional[autoextract_poet.items.Product]`
Extract an item from a web page

autoextract_poet.pages.AutoExtractRealEstatePage

class AutoExtractRealEstatePage(*real_estate_data*:
 autoextract_poet.page_inputs.AutoExtractRealEstateData)
 Bases: web_poet.pages.ItemPage
 Real estate data from AutoExtract
<https://docs.zyte.com/automatic-extraction/real-estate.html>
__init__(*real_estate_data*: autoextract_poet.page_inputs.AutoExtractRealEstateData) → None
 Method generated by attrs for class AutoExtractRealEstatePage.

Methods

<code>__init__(real_estate_data)</code>	Method generated by attrs for class AutoExtractRealEstatePage.
<code>to_item()</code>	Extract an item from a web page

Attributes

<code>real_estate_data</code>

to_item() → Optional[autoextract_poet.items.RealEstate]
 Extract an item from a web page

autoextract_poet.pages.AutoExtractReviewsPage

class AutoExtractReviewsPage(*reviews_data*: autoextract_poet.page_inputs.AutoExtractReviewsData)
 Bases: web_poet.pages.ItemPage
 Reviews data from AutoExtract
<https://docs.zyte.com/automatic-extraction/review.html>
__init__(*reviews_data*: autoextract_poet.page_inputs.AutoExtractReviewsData) → None
 Method generated by attrs for class AutoExtractReviewsPage.

Methods

<code>__init__(reviews_data)</code>	Method generated by attrs for class AutoExtractReviewsPage.
<code>to_item()</code>	Extract an item from a web page

Attributes

reviews_data

to_item() → *Optional[autoextract_poet.items.Reviews]*
 Extract an item from a web page

autoextract_poet.pages.AutoExtractVehiclePage

class AutoExtractVehiclePage(*vehicle_data*: autoextract_poet.page_inputs.AutoExtractVehicleData)

Bases: *web_poet.pages.ItemPage*

Vehicle data from AutoExtract

<https://docs.zyte.com/automatic-extraction/vehicle.html>

__init__(*vehicle_data*: autoextract_poet.page_inputs.AutoExtractVehicleData) → None
 Method generated by attrs for class AutoExtractVehiclePage.

Methods

__init__ (<i>vehicle_data</i>)	Method generated by attrs for class AutoExtractVehiclePage.
to_item ()	Extract an item from a web page

Attributes

vehicle_data

to_item() → *Optional[autoextract_poet.items.Vehicle]*
 Extract an item from a web page

autoextract_poet.pages.AutoExtractWebPage

class AutoExtractWebPage(*response*: autoextract_poet.page_inputs.AutoExtractHtml)

Bases: *web_poet.pages.Injectable*, *web_poet.mixins.ResponseShortcutsMixin*

Base Page Object which requires *AutoExtractHtml* and provides XPath / CSS shortcuts.

Use this class as a base class for Page Objects which work on the browser HTML provided by AutoExtract.

__init__(*response*: autoextract_poet.page_inputs.AutoExtractHtml) → None
 Method generated by attrs for class AutoExtractWebPage.

Methods

<code>__init__(response)</code>	Method generated by attrs for class <code>AutoExtractWebPage</code> .
<code>css(query)</code>	Run a CSS query on a response, using <code>parsel.Selector</code> .
<code>urljoin(url)</code>	Convert url to absolute, taking in account url and baseurl of the response
<code>xpath(query, **kwargs)</code>	Run an XPath query on a response, using <code>parsel.Selector</code> .

Attributes

<code>base_url</code>	Return the base url of the given response
<code>html</code>	Shortcut to HTML Response's content.
<code>selector</code>	<code>parsel.Selector</code> instance for the HTML Response.
<code>url</code>	Shortcut to HTML Response's URL.
<code>response</code>	

2.1.5 autoextract_poet.util

Functions

<code>export(fn)</code>	Decorator that includes the decorated element into the <code>__all__</code> variable in the module.
<code>split_dict(dict, key_pred)</code>	Splits the dictionary in two.
<code>split_in_unknown_and_known_fields(data, item_cls)</code>	Return a pair of dicts.

autoextract_poet.util.export

`export(fn)`

Decorator that includes the decorated element into the `__all__` variable in the module. Useful to control what is imported when `import * from <module>` is used.

autoextract_poet.util.split_dict

`split_dict(dict: Dict, key_pred: Callable[[Any], Any]) → Tuple[Dict, Dict]`

Splits the dictionary in two. The first dict contains the records for which the key predicate is False and the second dict contains the rest

```
>>> split_dict({}, lambda k: False)
({}, {})
>>> split_dict(dict(a=1, b=2, c=3), lambda k: k != 'a')
({'a': 1}, {'b': 2, 'c': 3})
```

autoextract_poet.util.split_in_unknown_and_known_fields

split_in_unknown_and_known_fields(*data*: *Optional[dict]*, *item_cls*: *Type*) → *Tuple[Dict, Dict]*

Return a pair of dicts. The first one contains those elements not belonging to the attr class `item_cls`. The second one contains the rest. That is, those attributes not belonging to `item_cls` class

CONTRIBUTING

autoextract-poet is an open-source project. Your contribution is very welcome!

3.1 Issue Tracker

If you have a bug report, a new feature proposal or simply would like to make a question, please check our issue tracker on Github: <https://github.com/scrapinghub/autoextract-poet/issues>

3.2 Source code

Our source code is hosted on Github: <https://github.com/scrapinghub/autoextract-poet>

Before opening a pull request, it might be worth checking current and previous issues. Some code changes might also require some discussion before being accepted so it might be worth opening a new issue before implementing huge or breaking changes.

3.3 Testing

We use `tox` to run tests with different Python versions:

```
tox
```

The command above also runs type checks; we use mypy.

CHANGELOG

4.1 x.y.z (YYYY-MM-DD)

- removed support for Python 3.6
- added support for Python 3.10
- (internal) Setup pre-commit linters like `black`, `isort`, and `flake8` to standardize code quality.

4.2 0.3.1 / 0.4.0 (2021-10-26)

- These two versions are the same and was uploaded as duplicates to PyPI.
- Supports new fields from AutoExtract for Product: `color`, `size`, `style`.

4.3 0.3.0 (2021-08-04)

- Support for all API page types at the moment
- Introduction of `_unknown_fields_dict` and `AutoExtractAdapter`. Allows to extend items with custom attributes and to include in the output the returned attributes not yet supported by the existing definitions.
- Initial documentation

4.4 0.2.2 (2021-05-29)

- Page classes for Article, Product and ProductList introduced

4.5 0.2.1 (2021-01-27)

- `AdditionalProperty` value as optional to match `unified-schema`

4.6 0.2.0 (2020-12-30)

- `AutoExtractProductListData` page input and `ProductList` item
- `from_dict` of items no longer fail on unknown attributes, they're ignored now
- List attributes now default to `[]` instead of `None`
- CI is switched to github actions
- Python 3.9 is added to CI

4.7 0.1.0 (2020-11-19)

- `AutoExtractHtml` page input
- `AutoExtractWebPage` and `AutoExtractItemWebPage` base page objects

4.8 0.0.1 (2020-08-18)

Initial release.

- Article and Product page inputs
- Article and Product items (and their dependencies)

LICENSE

Copyright (c) Zyte Group Ltd All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of Zyte nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PYTHON MODULE INDEX

a

- `autoextract_poet`, 5
- `autoextract_poet.adapters`, 5
- `autoextract_poet.items`, 7
- `autoextract_poet.page_inputs`, 35
- `autoextract_poet.pages`, 43
- `autoextract_poet.util`, 51

Symbols

[__init__\(\)](#) (*AdditionalProperty* method), 8
[__init__\(\)](#) (*Address* method), 9
[__init__\(\)](#) (*Area* method), 10
[__init__\(\)](#) (*Article* method), 10
[__init__\(\)](#) (*ArticleFromList* method), 12
[__init__\(\)](#) (*ArticleList* method), 13
[__init__\(\)](#) (*AutoExtractAdapter* method), 6
[__init__\(\)](#) (*AutoExtractArticleData* method), 36
[__init__\(\)](#) (*AutoExtractArticleListData* method), 37
[__init__\(\)](#) (*AutoExtractArticleListPage* method), 44
[__init__\(\)](#) (*AutoExtractArticlePage* method), 44
[__init__\(\)](#) (*AutoExtractCommentsData* method), 37
[__init__\(\)](#) (*AutoExtractCommentsPage* method), 45
[__init__\(\)](#) (*AutoExtractData* method), 38
[__init__\(\)](#) (*AutoExtractForumPostsData* method), 39
[__init__\(\)](#) (*AutoExtractForumPostsPage* method), 46
[__init__\(\)](#) (*AutoExtractHtml* method), 39
[__init__\(\)](#) (*AutoExtractItemWebPage* method), 46
[__init__\(\)](#) (*AutoExtractJobPostingData* method), 40
[__init__\(\)](#) (*AutoExtractJobPostingPage* method), 47
[__init__\(\)](#) (*AutoExtractProductData* method), 40
[__init__\(\)](#) (*AutoExtractProductListData* method), 41
[__init__\(\)](#) (*AutoExtractProductListPage* method), 47
[__init__\(\)](#) (*AutoExtractProductPage* method), 48
[__init__\(\)](#) (*AutoExtractRealEstateData* method), 42
[__init__\(\)](#) (*AutoExtractRealEstatePage* method), 49
[__init__\(\)](#) (*AutoExtractReviewsData* method), 42
[__init__\(\)](#) (*AutoExtractReviewsPage* method), 49
[__init__\(\)](#) (*AutoExtractVehicleData* method), 43
[__init__\(\)](#) (*AutoExtractVehiclePage* method), 50
[__init__\(\)](#) (*AutoExtractWebPage* method), 50
[__init__\(\)](#) (*AvailableAtOrFrom* method), 14
[__init__\(\)](#) (*Breadcrumb* method), 14
[__init__\(\)](#) (*Comment* method), 15
[__init__\(\)](#) (*Comments* method), 15
[__init__\(\)](#) (*ForumPost* method), 16
[__init__\(\)](#) (*ForumPosts* method), 17
[__init__\(\)](#) (*FuelEfficiency* method), 18
[__init__\(\)](#) (*GTIN* method), 18
[__init__\(\)](#) (*Item* method), 19
[__init__\(\)](#) (*JobPosting* method), 19

[__init__\(\)](#) (*Location* method), 20
[__init__\(\)](#) (*MileageFromOdometer* method), 21
[__init__\(\)](#) (*Offer* method), 21
[__init__\(\)](#) (*Organization* method), 22
[__init__\(\)](#) (*PaginationLink* method), 23
[__init__\(\)](#) (*Product* method), 23
[__init__\(\)](#) (*ProductFromList* method), 25
[__init__\(\)](#) (*ProductList* method), 26
[__init__\(\)](#) (*Rating* method), 27
[__init__\(\)](#) (*RealEstate* method), 27
[__init__\(\)](#) (*Review* method), 29
[__init__\(\)](#) (*Reviews* method), 30
[__init__\(\)](#) (*Salary* method), 31
[__init__\(\)](#) (*Topic* method), 32
[__init__\(\)](#) (*TradeAction* method), 32
[__init__\(\)](#) (*Vehicle* method), 33
[__init__\(\)](#) (*VehicleEngine* method), 35

A

[AdditionalProperty](#) (class in *autoextract_poet.items*), 8
[Address](#) (class in *autoextract_poet.items*), 9
[Area](#) (class in *autoextract_poet.items*), 10
[Article](#) (class in *autoextract_poet.items*), 10
[ArticleFromList](#) (class in *autoextract_poet.items*), 12
[ArticleList](#) (class in *autoextract_poet.items*), 13
[autoextract_poet](#)
 module, 5
[autoextract_poet.adapters](#)
 module, 5
[autoextract_poet.items](#)
 module, 7
[autoextract_poet.page_inputs](#)
 module, 35
[autoextract_poet.pages](#)
 module, 43
[autoextract_poet.util](#)
 module, 51
[AutoExtractAdapter](#) (class in *autoextract_poet.adapters*), 6
[AutoExtractArticleData](#) (class in *autoextract_poet.page_inputs*), 36

[AutoExtractArticleListData](#) (class in [autoextract_poet.page_inputs](#)), 37
[AutoExtractArticleListPage](#) (class in [autoextract_poet.pages](#)), 44
[AutoExtractArticlePage](#) (class in [autoextract_poet.pages](#)), 44
[AutoExtractCommentsData](#) (class in [autoextract_poet.page_inputs](#)), 37
[AutoExtractCommentsPage](#) (class in [autoextract_poet.pages](#)), 45
[AutoExtractData](#) (class in [autoextract_poet.page_inputs](#)), 38
[AutoExtractForumPostsData](#) (class in [autoextract_poet.page_inputs](#)), 39
[AutoExtractForumPostsPage](#) (class in [autoextract_poet.pages](#)), 46
[AutoExtractHtml](#) (class in [autoextract_poet.page_inputs](#)), 39
[AutoExtractItemWebPage](#) (class in [autoextract_poet.pages](#)), 46
[AutoExtractJobPostingData](#) (class in [autoextract_poet.page_inputs](#)), 40
[AutoExtractJobPostingPage](#) (class in [autoextract_poet.pages](#)), 47
[AutoExtractProductData](#) (class in [autoextract_poet.page_inputs](#)), 40
[AutoExtractProductListData](#) (class in [autoextract_poet.page_inputs](#)), 41
[AutoExtractProductListPage](#) (class in [autoextract_poet.pages](#)), 47
[AutoExtractProductPage](#) (class in [autoextract_poet.pages](#)), 48
[AutoExtractRealEstateData](#) (class in [autoextract_poet.page_inputs](#)), 42
[AutoExtractRealEstatePage](#) (class in [autoextract_poet.pages](#)), 49
[AutoExtractReviewsData](#) (class in [autoextract_poet.page_inputs](#)), 42
[AutoExtractReviewsPage](#) (class in [autoextract_poet.pages](#)), 49
[AutoExtractVehicleData](#) (class in [autoextract_poet.page_inputs](#)), 43
[AutoExtractVehiclePage](#) (class in [autoextract_poet.pages](#)), 50
[AutoExtractWebPage](#) (class in [autoextract_poet.pages](#)), 50
[AvailableAtOrFrom](#) (class in [autoextract_poet.items](#)), 14

B

[Breadcrumb](#) (class in [autoextract_poet.items](#)), 14

C

[Comment](#) (class in [autoextract_poet.items](#)), 15

[Comments](#) (class in [autoextract_poet.items](#)), 15

E

[export\(\)](#) (in module [autoextract_poet.util](#)), 51

F

[field_names\(\)](#) ([AutoExtractAdapter](#) method), 7
[ForumPost](#) (class in [autoextract_poet.items](#)), 16
[ForumPosts](#) (class in [autoextract_poet.items](#)), 17
[from_dict\(\)](#) ([Article](#) class method), 11
[from_dict\(\)](#) ([ArticleList](#) class method), 13
[from_dict\(\)](#) ([Comments](#) class method), 16
[from_dict\(\)](#) ([ForumPosts](#) class method), 17
[from_dict\(\)](#) ([Item](#) class method), 19
[from_dict\(\)](#) ([JobPosting](#) class method), 20
[from_dict\(\)](#) ([Product](#) class method), 24
[from_dict\(\)](#) ([ProductFromList](#) class method), 25
[from_dict\(\)](#) ([ProductList](#) class method), 26
[from_dict\(\)](#) ([RealEstate](#) class method), 29
[from_dict\(\)](#) ([Review](#) class method), 30
[from_dict\(\)](#) ([Reviews](#) class method), 31
[from_dict\(\)](#) ([Vehicle](#) class method), 35
[from_list\(\)](#) ([Item](#) class method), 19
[FuelEfficiency](#) (class in [autoextract_poet.items](#)), 18

G

[get_field_meta\(\)](#) ([AutoExtractAdapter](#) method), 7
[get_item_class\(\)](#) (in module [autoextract_poet.page_inputs](#)), 36
[GTIN](#) (class in [autoextract_poet.items](#)), 18

I

[is_item\(\)](#) ([AutoExtractAdapter](#) class method), 7
[Item](#) (class in [autoextract_poet.items](#)), 19

J

[JobPosting](#) (class in [autoextract_poet.items](#)), 19

L

[Location](#) (class in [autoextract_poet.items](#)), 20

M

[MileageFromOdometer](#) (class in [autoextract_poet.items](#)), 21
 module
 [autoextract_poet](#), 5
 [autoextract_poet.adapters](#), 5
 [autoextract_poet.items](#), 7
 [autoextract_poet.page_inputs](#), 35
 [autoextract_poet.pages](#), 43
 [autoextract_poet.util](#), 51

O

Offer (class in *autoextract_poet.items*), 21

Organization (class in *autoextract_poet.items*), 22

P

PaginationLink (class in *autoextract_poet.items*), 23

Product (class in *autoextract_poet.items*), 23

ProductFromList (class in *autoextract_poet.items*), 25

ProductList (class in *autoextract_poet.items*), 26

R

Rating (class in *autoextract_poet.items*), 27

RealEstate (class in *autoextract_poet.items*), 27

Review (class in *autoextract_poet.items*), 29

Reviews (class in *autoextract_poet.items*), 30

S

Salary (class in *autoextract_poet.items*), 31

split_dict() (in module *autoextract_poet.util*), 51

split_in_unknown_and_known_fields() (in module *autoextract_poet.util*), 52

T

to_item() (*AutoExtractArticleListPage* method), 44

to_item() (*AutoExtractArticlePage* method), 45

to_item() (*AutoExtractCommentsPage* method), 45

to_item() (*AutoExtractForumPostsPage* method), 46

to_item() (*AutoExtractJobPostingPage* method), 47

to_item() (*AutoExtractProductListPage* method), 48

to_item() (*AutoExtractProductPage* method), 48

to_item() (*AutoExtractRealEstatePage* method), 49

to_item() (*AutoExtractReviewsPage* method), 50

to_item() (*AutoExtractVehiclePage* method), 50

Topic (class in *autoextract_poet.items*), 32

TradeAction (class in *autoextract_poet.items*), 32

V

Vehicle (class in *autoextract_poet.items*), 33

VehicleEngine (class in *autoextract_poet.items*), 35