autoextract-poet Documentation

Release 0.3.1

Zyte

Dec 03, 2021

GETTING STARTED

`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.

CHAPTER

ONE

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 `attr.s` 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.

CHAPTER

TWO

API REFERENCE

2.1 autoextract_poet

Modules

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

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

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

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

__init__(name[, value])	Method generated by attrs for class AdditionalProperty.
from_dict(item)	Read an item from a dictionary.
from_list(items)	Read items from a list, invoking from_dict 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

__init__([postalCode, streetAddress, ...])	Method generated by attrs for class Address.
from_dict(item)	Read an item from a dictionary.
from_list(items)	Read items from a list, invoking from_dict 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

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

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

Attributes

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

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

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

Attributes

url
probability
headline
datePublished
datePublishedRaw
author
authorsList
inLanguage
mainImage

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

__init__([url, articles, paginationNext, ...])	Method generated by attrs for class ArticleList.
from_dict(item)	Read an item from a dictionary.
from_list(items)	Read items from a list, invoking from_dict 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

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

Attributes

raw

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

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

Attributes

name
link

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

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

Attributes

probability
text
datePublished
datePublishedRaw
upvoteCount
downvoteCount

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

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

Attributes

url
comments

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

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

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

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

Attributes

raw

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

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

Attributes

type
value

autoextract_poet.items.Item

class Item

Bases: autoextract_poet.items._ItemBase

__init__() → None

Method generated by attrs for class Item.

Methods

__init__()	Method generated by attrs for class Item.
from_dict(item)	Read an item from a dictionary.
from_list(items)	Read items from a list, invoking from_dict 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

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

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

Attributes

`raw`

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

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

Attributes

`value`
`unitCode`

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

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

Attributes

price
currency
availability
regularPrice

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

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

Attributes

raw

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

__init__([url, text])	Method generated by attrs for class PaginationLink.
from_dict(item)	Read an item from a dictionary.
from_list(items)	Read items from a list, invoking from_dict 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

__init__([url, canonicalUrl, probability, ...])	Method generated by attrs for class Product.
from_dict(item)	Read an item from a dictionary.
from_list(items)	Read items from a list, invoking from_dict 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

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

Attributes

probability
url
name
offers
sku
brand
mainImage
images
description
aggregateRating

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

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

Attributes

url
products
breadcrumbs
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.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

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

Attributes

ratingValue
bestRating
reviewCount

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

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

Attributes

url
probability
name
datePublished
datePublishedRaw
description
mainImage
images
yearBuilt
breadcrumbs
additionalProperty
address
area

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

__init__([name, reviewBody, reviewRating, ...])	Method generated by attrs for class Review.
from_dict(item)	Read an item from a dictionary.
from_list(items)	Read items from a list, invoking `from_dict` 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

__init__([url, reviews, paginationNext, ...])	Method generated by attrs for class Reviews.
from_dict(item)	Read an item from a dictionary.
from_list(items)	Read items from a list, invoking `from_dict` 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

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

__init__([name])	Method generated by attrs for class Topic.
from_dict(item)	Read an item from a dictionary.
from_list(items)	Read items from a list, invoking from_dict 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

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

__init__([url, canonicalUrl, probability, ...])	Method generated by attrs for class Vehicle.
from_dict(item)	Read an item from a dictionary.
from_list(items)	Read items from a list, invoking from_dict 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

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

Attributes

raw

2.1.3 autoextract_poet.page_inputs

Functions

get_item_class(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

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

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

Attributes

item_class
page_type

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

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

Attributes

item_class
page_type
data

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__(data)	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__(url, html)	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

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

Attributes

item_class
page_type

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

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

Attributes

item_class
page_type

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

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

Attributes

item_class
page_type

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

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

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

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

continues on next page

Table 98 – continued from previous page

AutoExtractReviewsPage(reviews_data)	Reviews data from AutoExtract
AutoExtractVehiclePage(vehicle_data)	Vehicle data from AutoExtract
AutoExtractWebPage(response)	Base Page Object which requires *AutoExtractHtml* 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

__init__(article_list_data)	Method generated by attrs for class AutoExtractArticleListPage.
to_item()	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

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

Attributes

article_data

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

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

Attributes

comments_data

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

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

Attributes

forum_posts_data

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

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

Attributes

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

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

Attributes

job_posting_data

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

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

Attributes

product_list_data

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

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

Attributes

product_data

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

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

Attributes

real_estate_data

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

__init__(reviews_data)	Method generated by attrs for class AutoExtractReviewsPage.
to_item()	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__(vehicle_data)	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

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

Attributes

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

2.1.5 autoextract_poet.util

Functions

export(fn)	Decorator that includes the decorated element into the __all__ variable in the module.
split_dict(dict, key_pred)	Splits the dictionary in two.
split_in_unknown_and_known_fields(data, item_cls)	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

CHAPTER

THREE

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.

CHAPTER

FOUR

CHANGELOG

4.1 0.3.1 (2021-10-26)

- Supports new fields from AutoExtract for Product: `color`, `size`, `style`.

4.2 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

4.3 0.2.2 (2021-05-29)

- Page classes for Article, Product and ProductList introduced

4.4 0.2.1 (2021-01-27)

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

4.5 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

4.6 0.1.0 (2020-11-19)

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

4.7 0.0.1 (2020-08-18)

Initial release.

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

CHAPTER

FIVE

LICENSE

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.

PYTHON MODULE INDEX

a

INDEX

Symbols

A

B

C

E

F

G

I

J

L

M

O

P

R

S

T

V