
Prismic Documentation

Release 1.0.2

Prismic.io

Aug 30, 2017

Contents

1	prismic Package	3
1.1	prismic Package	3
1.2	api Module	3
1.3	core Module	6
1.4	exceptions Module	6
1.5	fragments Module	7
1.6	structured_text Module	11
2	Indices and tables	13
	Python Module Index	15

Python development kit for prismic.io

Example of use:

```
>>> import prismic
>>> api = prismic.get("http://lesbonneschoses.prismic.io/api", "")
>>> form = api.form("everything")

>>> # Set the ref and the query to get all documents of type "product"
>>> form.ref(api.get_master()).query("""[:d = any(document.type, ["product"])]""")

>>> documents = form.submit().documents
>>> documents[0].get_text("product.name")
u'Speculoos Macaron'
```

Prismic API:

prismic Package

Prismic.io python library.

api Module

prismic.api

This module implements the Prismic API.

class `prismic.api.Api` (*data, access_token, cache, request_handler*)
Bases: `object`

A Prismic API, pointing to a specific repository. Use `prismic.api.get()` to fetch one.

Variables

- **bookmarks** (*dict*) – all bookmarks, as a dict from name to document id
- **types** (*array<str>*) – all available types
- **tags** (*array<str>*) – all available tags
- **experiments** (*Experiments*) – information about current experiments
- **access_token** (*str*) – current access token (may be None)

form (*name*)

Constructs the form with data from Api. Returns *SearchForm* object.

Parameters *name* – Name of the form.

get_by_id (*id, ref=None*)

get_by_ids (*ids, ref=None, page_size=None, page=None, orderings=None, after=None, fetch_links=None*)

get_by_uid (*type, uid, ref=None*)

get_master ()

Returns current master *Ref* object.

get_ref (*label*)

Get the *Ref* with a specific label. Returns *Ref* object.

Parameters *label* – Name of the label.

get_single (*type, ref=None*)

preview_session (*token, link_resolver, default_url*)

Return the URL to display a given preview

:param *token* as received from Prismic server to identify the content to preview :param *link_resolver* the link resolver to build URL for your site :param *default_url* the URL to default to return if the preview doesn't correspond to a document

(usually the home page of your site)

Returns the URL to redirect the user to

query (*q, ref=None, page_size=None, page=None, orderings=None, after=None, fetch_links=None*)

query_first (*q, ref=None*)

class `prismic.api.Document` (*data*)

Bases: `prismic.fragments.WithFragments`

Represents a Prismic.io Document

Variables

- **id** (*str*) – document id
- **uid** (*str*) – document uid
- **type** (*str*) –
- **href** (*str*) –
- **tags** (*array<str>*) –
- **slugs** (*array<str>*) –

as_link ()

Convert the current document to a `DocumentLink`

Returns `DocumentLink`

slug

Return the most recent slug

Returns `str` slug

class `prismic.api.Ref` (*data*)

Bases: `object`

A Prismic.io Reference (corresponds to a release)

class `prismic.api.Response` (*data*)

Bases: `object`

Prismic's response to a query.

Variables

- **documents** (*array<prismic.api.Document>*) – the documents of the current page
- **page** (*int*) – the page in this result, starting by 1
- **results_per_page** (*int*) – max result in a page
- **total_results_size** (*int*) – total number of results for this query
- **total_pages** (*int*) – total number of pages for this query
- **next_page** (*str*) – URL of the next page (may be None if on the last page)
- **prev_page** (*str*) – URL of the previous page (may be None)
- **results_size** (*int*) – number of results actually returned for the current page

class `prismic.api.SearchForm` (*form, access_token, cache, request_handler*)

Bases: `object`

Form to search for documents. Most of the methods return self object to allow chaining.

after (*doc_id*)

Start the result set after the given id

Parameters **doc_id** – id of the reference document

count ()

Count the total number of results

fetch (*fields*)

Restrict the results document to the specified fields

Parameters **fields** – The list of fields, array or comma separated string

fetch_links (*fields*)

Include the requested fields in the DocumentLink instances in the result

Parameters **fields** – The list of fields, array or comma separated string

orderings (*orderings*)

Sets the query orderings

:param orderings String with the orderings predicate :returns: the SearchForm instance to chain calls

page (*page_number*)

Set query page number

Parameters **page_number** – int representing the page number

pageSize (*nb_results*)

Deprecated: use page_size instead

page_size (*nb_results*)

Set query page size

Parameters **nb_results** – int representing the number of results per page

query (**argv*)

Parameters **argv** – Either a string query, or any number of Array corresponding to predicates.

See the `prismic.predicates` module for helper functions.

ref (*ref*)

Parameters **ref** – A `Ref` object or an string.

set (*field, value*)

submit ()

Submit the query to the Prismic.io server

Returns *Response*

submit_assert_preconditions ()

`prismic.api.get (url, access_token=None, cache=None, request_handler=None)`

Fetches the prismic api JSON. Returns *Api* object.

Parameters

- **url** – URL to the api of the repository (mandatory).
- **access_token** – The access token (optional).
- **cache** – The cache object. Optional, will default to a file-based cache if None is passed.
- **request_handler** – The request handler. Optional, will default to a request handler based on requests module.

core Module

exceptions Module

prismic.exceptions

This module contains the set of Prismic exceptions.

exception `prismic.exceptions.AuthorizationNeededError`

Bases: *prismic.exceptions.Error*

You need to provide an access token to access this repository

exception `prismic.exceptions.Error`

Bases: `exceptions.Exception`

Base class for exceptions in this module.

exception `prismic.exceptions.HTTPError` (*code, message*)

Bases: *prismic.exceptions.Error*

HTTP error

exception `prismic.exceptions.InvalidTokenError`

Bases: *prismic.exceptions.Error*

The provided access token is either invalid or expired

exception `prismic.exceptions.InvalidURLError`

Bases: *prismic.exceptions.Error*

The provider URL is invalid

exception `prismic.exceptions.RefMissing`

Bases: *prismic.exceptions.Error*

You need to provide the ref parameter.

exception `prismic.exceptions.UnexpectedError` (*message*)
 Bases: `prismic.exceptions.Error`
 Unexpected error.

fragments Module

class `prismic.fragments.Block`
 Bases: `object`
 A block in a structured text

class `Embed` (*embed*)
 Bases: `object`
get_embed()

class `Block.Heading` (*value*)
 Bases: `prismic.fragments.Text`

class `Block.Image` (*view*)
 Bases: `object`
 Block image
Parameters **view** – The `Fragment.Image.View` object
get_view()

class `Block.ListItem` (*value, is_ordered=False*)
 Bases: `prismic.fragments.Text`

class `Block.Paragraph` (*value*)
 Bases: `prismic.fragments.Text`

class `prismic.fragments.Fragment`
 Bases: `object`

class `BasicFragment` (*value*)
 Bases: `prismic.fragments.FragmentElement`

class `Fragment.Color` (*value*)
 Bases: `prismic.fragments.BasicFragment`
as_html

class `Fragment.Date` (*value*)
 Bases: `prismic.fragments.BasicFragment`
as_datetime
as_html

class `Fragment.DocumentLink` (*value*)
 Bases: `prismic.fragments.WithFragments, prismic.fragments.Link`
as_html (*documentlink_resolver, html_serializer=None*)
 Get the `DocumentLink` as html.
Parameters **documentlink_resolver** – A resolver function will be called with `prismic.fragments.Fragment.DocumentLink` object as argument. Resolver function should return a string, the local url to the document.
get_document_id()

```
    get_document_slug()
    get_document_tags()
    get_document_type()
    get_url(documentlink_resolver=None)
class Fragment.Embed(value)
    Bases: prismic.fragments.FragmentElement
    as_html
class Fragment.FileLink(value)
    Bases: prismic.fragments.Link
    as_html()
    get_file()
    get_filename()
    get_url(link_resolver=None)
class Fragment.GeoPoint(value)
    Bases: prismic.fragments.FragmentElement
    as_html
class Fragment.Group(value)
    Bases: prismic.fragments.BasicFragment
    as_html(link_resolver)
class Fragment.Image(value)
    Bases: prismic.fragments.FragmentElement
    class View(data)
        Bases: prismic.fragments.FragmentElement
        View class
        as_html(link_resolver)
        ratio

    Fragment.Image.as_html(link_resolver)
    Fragment.Image.get_view(key)
class Fragment.ImageLink(value)
    Bases: prismic.fragments.Link
    as_html
    get_image()
    get_url()
class Fragment.Link
    Bases: prismic.fragments.FragmentElement
    static parse(data)
class Fragment.MediaLink(value)
    Bases: prismic.fragments.Link
    as_html()
```

```

    get_url (link_resolver=None)

class Fragment.Number (value)
    Bases: prismic.fragments.BasicFragment

    as_html

class Fragment.Range (value)
    Bases: prismic.fragments.BasicFragment

    as_html

class Fragment.Slice (slice_type, slice_label, value)
    Bases: prismic.fragments.FragmentElement

    as_html (link_resolver)

class Fragment.SliceZone (value)
    Bases: prismic.fragments.FragmentElement

    as_html (link_resolver)

class Fragment.Text (value)
    Bases: prismic.fragments.BasicFragment

    as_html

class Fragment.Timestamp (value)
    Bases: prismic.fragments.BasicFragment

    as_datetime

    as_html

class Fragment.WebLink (value)
    Bases: prismic.fragments.Link

    as_html

    get_url (link_resolver=None)

class Fragment.WithFragments (fragments)
    Bases: object

    as_html (link_resolver)

    static fragment_to_html (fragment, link_resolver, html_serializer=None)

    get (field)

    get_all (field)

    get_color (field)

    get_date (field)

    get_embed (field)

    get_field (field)

    get_fragment_type (field, f_type)

    get_geopoint (field)

    get_group (field)

    get_html (field, link_resolver)
        Get the html of a field.

```

Parameters

- **field** – String with a name of the field to get.
- **link_resolver** – A resolver function for document links.

Will be called with `prismic.fragments.Fragment.DocumentLink` object as argument.

Resolver function should return a string, the local url to the document.

get_image (*field*, *view=u'main'*)

get_link (*field*)

get_number (*field*)

get_range (*field*)

get_slice_zone (*field*)

get_structured_text (*field*)

get_text (*field*)

get_timestamp (*field*)

linked_documents

Return the documents linked from this document's fragments :return: array<DocumentLink>

classmethod `Fragment.from_json` (*data*)

Create a corresponding fragment object from json.

class `prismic.fragments.FragmentElement`

Bases: object

class `prismic.fragments.Span`

Bases: object

class `Em` (*value*)

Bases: `prismic.fragments.SpanElement`

class `Span.Hyperlink` (*value*)

Bases: `prismic.fragments.SpanElement`

get_url (*link_resolver*)

class `Span.SpanElement` (*value*)

Bases: object

length ()

class `Span.Strong` (*value*)

Bases: `prismic.fragments.SpanElement`

classmethod `Span.from_json` (*data*)

class `prismic.fragments.StructuredText` (*values*)

Bases: object

class `Group` (*tag*, *blocks*)

Bases: object

`StructuredText.as_html` (*link_resolver*, *html_serializer=None*)

static `StructuredText.block_as_html` (*block*, *content*, *link_resolver*, *html_serializer*)

`StructuredText.get_first_paragraph` ()

`StructuredText.get_image` ()

`StructuredText.get_title` ()

```
static StructuredText.span_as_html (text, spans, link_resolver, html_serializer)
```

```
static StructuredText.span_write_tag (span, content, link_resolver, html_serializer)
```

```
class prismic.fragments.Text (value)
```

```
    Bases: object
```

```
    Base class for blocks
```

structured_text Module

To activate debugging:

```
>>> import logging
>>> logging.basicConfig(level=logging.DEBUG)
```


CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

p

- `prismic.__init__`, 3
- `prismic.api`, 3
- `prismic.exceptions`, 6
- `prismic.fragments`, 7

A

`after()` (prismic.api.SearchForm method), 5
`Api` (class in prismic.api), 3
`as_datetime` (prismic.fragments.Fragment.Date attribute), 7
`as_datetime` (prismic.fragments.Fragment.Timestamp attribute), 9
`as_html` (prismic.fragments.Fragment.Color attribute), 7
`as_html` (prismic.fragments.Fragment.Date attribute), 7
`as_html` (prismic.fragments.Fragment.Embed attribute), 8
`as_html` (prismic.fragments.Fragment.GeoPoint attribute), 8
`as_html` (prismic.fragments.Fragment.ImageLink attribute), 8
`as_html` (prismic.fragments.Fragment.Number attribute), 9
`as_html` (prismic.fragments.Fragment.Range attribute), 9
`as_html` (prismic.fragments.Fragment.Text attribute), 9
`as_html` (prismic.fragments.Fragment.Timestamp attribute), 9
`as_html` (prismic.fragments.Fragment.WebLink attribute), 9
`as_html()` (prismic.fragments.Fragment.DocumentLink method), 7
`as_html()` (prismic.fragments.Fragment.FileLink method), 8
`as_html()` (prismic.fragments.Fragment.Group method), 8
`as_html()` (prismic.fragments.Fragment.Image method), 8
`as_html()` (prismic.fragments.Fragment.Image.View method), 8
`as_html()` (prismic.fragments.Fragment.MediaLink method), 8
`as_html()` (prismic.fragments.Fragment.Slice method), 9
`as_html()` (prismic.fragments.Fragment.SliceZone method), 9
`as_html()` (prismic.fragments.Fragment.WithFragments method), 9
`as_html()` (prismic.fragments.StructuredText method), 10
`as_link()` (prismic.api.Document method), 4

`AuthorizationNeededError`, 6

B

`Block` (class in prismic.fragments), 7
`Block.Embed` (class in prismic.fragments), 7
`Block.Heading` (class in prismic.fragments), 7
`Block.Image` (class in prismic.fragments), 7
`Block.ListItem` (class in prismic.fragments), 7
`Block.Paragraph` (class in prismic.fragments), 7
`block_as_html()` (prismic.fragments.StructuredText static method), 10

C

`count()` (prismic.api.SearchForm method), 5

D

`Document` (class in prismic.api), 4

E

`Error`, 6

F

`fetch()` (prismic.api.SearchForm method), 5
`fetch_links()` (prismic.api.SearchForm method), 5
`form()` (prismic.api.Api method), 3
`Fragment` (class in prismic.fragments), 7
`Fragment.BasicFragment` (class in prismic.fragments), 7
`Fragment.Color` (class in prismic.fragments), 7
`Fragment.Date` (class in prismic.fragments), 7
`Fragment.DocumentLink` (class in prismic.fragments), 7
`Fragment.Embed` (class in prismic.fragments), 8
`Fragment.FileLink` (class in prismic.fragments), 8
`Fragment.GeoPoint` (class in prismic.fragments), 8
`Fragment.Group` (class in prismic.fragments), 8
`Fragment.Image` (class in prismic.fragments), 8
`Fragment.Image.View` (class in prismic.fragments), 8
`Fragment.ImageLink` (class in prismic.fragments), 8
`Fragment.Link` (class in prismic.fragments), 8
`Fragment.MediaLink` (class in prismic.fragments), 8

Fragment.Number (class in prismic.fragments), 9
Fragment.Range (class in prismic.fragments), 9
Fragment.Slice (class in prismic.fragments), 9
Fragment.SliceZone (class in prismic.fragments), 9
Fragment.Text (class in prismic.fragments), 9
Fragment.Timestamp (class in prismic.fragments), 9
Fragment.WebLink (class in prismic.fragments), 9
Fragment.WithFragments (class in prismic.fragments), 9
fragment_to_html() (prismic.fragments.Fragment.WithFragments static method), 9
FragmentElement (class in prismic.fragments), 10
from_json() (prismic.fragments.Fragment class method), 10
from_json() (prismic.fragments.Span class method), 10

G

get() (in module prismic.api), 6
get() (prismic.fragments.Fragment.WithFragments method), 9
get_all() (prismic.fragments.Fragment.WithFragments method), 9
get_by_id() (prismic.api.Api method), 3
get_by_ids() (prismic.api.Api method), 3
get_by_uid() (prismic.api.Api method), 4
get_color() (prismic.fragments.Fragment.WithFragments method), 9
get_date() (prismic.fragments.Fragment.WithFragments method), 9
get_document_id() (prismic.fragments.Fragment.DocumentLink method), 7
get_document_slug() (prismic.fragments.Fragment.DocumentLink method), 7
get_document_tags() (prismic.fragments.Fragment.DocumentLink method), 8
get_document_type() (prismic.fragments.Fragment.DocumentLink method), 8
get_embed() (prismic.fragments.Block.Embed method), 7
get_embed() (prismic.fragments.Fragment.WithFragments method), 9
get_field() (prismic.fragments.Fragment.WithFragments method), 9
get_file() (prismic.fragments.Fragment.FileLink method), 8
get_filename() (prismic.fragments.Fragment.FileLink method), 8
get_first_paragraph() (prismic.fragments.StructuredText method), 10

get_fragment_type() (prismic.fragments.Fragment.WithFragments method), 9
get_geopoint() (prismic.fragments.Fragment.WithFragments method), 9
get_group() (prismic.fragments.Fragment.WithFragments method), 9
get_html() (prismic.fragments.Fragment.WithFragments method), 9
get_image() (prismic.fragments.Fragment.ImageLink method), 8
get_image() (prismic.fragments.Fragment.WithFragments method), 10
get_image() (prismic.fragments.StructuredText method), 10
get_link() (prismic.fragments.Fragment.WithFragments method), 10
get_master() (prismic.api.Api method), 4
get_number() (prismic.fragments.Fragment.WithFragments method), 10
get_range() (prismic.fragments.Fragment.WithFragments method), 10
get_ref() (prismic.api.Api method), 4
get_single() (prismic.api.Api method), 4
get_slice_zone() (prismic.fragments.Fragment.WithFragments method), 10
get_structured_text() (prismic.fragments.Fragment.WithFragments method), 10
get_text() (prismic.fragments.Fragment.WithFragments method), 10
get_timestamp() (prismic.fragments.Fragment.WithFragments method), 10
get_title() (prismic.fragments.StructuredText method), 10
get_url() (prismic.fragments.Fragment.DocumentLink method), 8
get_url() (prismic.fragments.Fragment.FileLink method), 8
get_url() (prismic.fragments.Fragment.ImageLink method), 8
get_url() (prismic.fragments.Fragment.MediaLink method), 8
get_url() (prismic.fragments.Fragment.WebLink method), 9
get_url() (prismic.fragments.Span.Hyperlink method), 10
get_view() (prismic.fragments.Block.Image method), 7
get_view() (prismic.fragments.Fragment.Image method), 8

H

HTTPError, 6

I

InvalidTokenError, 6

[InvalidURLError](#), 6

L

[length\(\)](#) ([prismic.fragments.Span.SpanElement](#) method), 10

[linked_documents](#) ([prismic.fragments.Fragment.WithFragments](#) attribute), 10

O

[orderings\(\)](#) ([prismic.api.SearchForm](#) method), 5

P

[page\(\)](#) ([prismic.api.SearchForm](#) method), 5

[page_size\(\)](#) ([prismic.api.SearchForm](#) method), 5

[pageSize\(\)](#) ([prismic.api.SearchForm](#) method), 5

[parse\(\)](#) ([prismic.fragments.Fragment.Link](#) static method), 8

[preview_session\(\)](#) ([prismic.api.Api](#) method), 4

[prismic.__init__](#) (module), 3

[prismic.api](#) (module), 3

[prismic.exceptions](#) (module), 6

[prismic.fragments](#) (module), 7

Q

[query\(\)](#) ([prismic.api.Api](#) method), 4

[query\(\)](#) ([prismic.api.SearchForm](#) method), 5

[query_first\(\)](#) ([prismic.api.Api](#) method), 4

R

[ratio](#) ([prismic.fragments.Fragment.Image.View](#) attribute), 8

[Ref](#) (class in [prismic.api](#)), 4

[ref\(\)](#) ([prismic.api.SearchForm](#) method), 5

[RefMissing](#), 6

[Response](#) (class in [prismic.api](#)), 4

S

[SearchForm](#) (class in [prismic.api](#)), 5

[set\(\)](#) ([prismic.api.SearchForm](#) method), 5

[slug](#) ([prismic.api.Document](#) attribute), 4

[Span](#) (class in [prismic.fragments](#)), 10

[Span.Em](#) (class in [prismic.fragments](#)), 10

[Span.Hyperlink](#) (class in [prismic.fragments](#)), 10

[Span.SpanElement](#) (class in [prismic.fragments](#)), 10

[Span.Strong](#) (class in [prismic.fragments](#)), 10

[span_as_html\(\)](#) ([prismic.fragments.StructuredText](#) static method), 10

[span_write_tag\(\)](#) ([prismic.fragments.StructuredText](#) static method), 11

[StructuredText](#) (class in [prismic.fragments](#)), 10

[StructuredText.Group](#) (class in [prismic.fragments](#)), 10

[submit\(\)](#) ([prismic.api.SearchForm](#) method), 6

[submit_assert_preconditions\(\)](#) ([prismic.api.SearchForm](#) method), 6

T

[Text](#) (class in [prismic.fragments](#)), 11

U

[UnexpectedError](#), 6