
django-bigbuild Documentation

Los Angeles Times Data Desk

May 31, 2019

Contents:

1	API	1
1.1	Context processors	1
1.2	Exceptions	1
1.3	Management commands	2
1.4	Models	2
	Python Module Index	5
	Index	7

1.1 Context processors

<i>bigbuild.context_processors.bigbuild</i>	Adds env-related context variables to the context.
---	--

`bigbuild.context_processors.bigbuild(request)`
 Adds env-related context variables to the context.

1.2 Exceptions

Custom exceptions and warnings

<i>bigbuild.exceptions.BadMetadata</i>	A custom exception to raise when the page configuration file has malformed metadata.
--	--

`bigbuild.exceptions.
MissingMetadataWarning`

<i>bigbuild.exceptions. MissingRecommendedMetadataWarning</i>	A custom warning that will be raised when a published page is missing recommended metadata values.
---	--

class `bigbuild.exceptions.BadMetadata(*args, **kwargs)`
 Bases: `exceptions.Exception`

A custom exception to raise when the page configuration file has malformed metadata.

class `bigbuild.exceptions.MissingRecommendedMetadataWarning(page)`
 Bases: `bigbuild.exceptions.BaseWarning`

A custom warning that will be raised when a published page is missing recommended metadata values.

For instance, things like a headline and description are not required to publish a page, but you're likely to have bugs with how your page appears on social media if you lack them.

`get_context_data()`

Returns the data to be passed into the message template as context.

1.3 Management commands

TK

1.4 Models

<code>bigbuild.models.Page</code>	A custom page published via static.latimes.com
<code>bigbuild.models.ArchivedPage</code>	An archived custom page.
<code>bigbuild.models.PageList</code>	A list of all the Page and ArchivedPage objects in the application.

class `bigbuild.models.Page(*args, **kwargs)`

Bases: `bigbuild.models.base.BasePage`

A custom page published via static.latimes.com

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

create_directory (*force=False, index_template_name=u'bigbuild/pages/default_index.html', index_template_context={}*)

Creates a new directory for the page.

Returns the path to the directory that has been created, which is the same as the `self.page_directory_path` property.

Throws an error if the directory already exists. You can force it to overwrite a pre-existing directory by submitting the `force` keyword argument as `true`.

delete ()

Delete the page directory.

frontmatter_path

Returns the `metadata.md` path where this page will be configured.

get_template (*name*)

Returns a Django template of the provided name ready to render.

write_checklist ()

Creates `checklist.md` in the page directory.

write_frontmatter (*path=None*)

Creates `metadata.yaml` in the page directory, or a supplied target directory.

write_index (*template_name, template_context={}*)

Creates `index.html` in the page directory.

write_static ()

Creates a `./static/` subdirectory within the page directory.

```
class bigbuild.models.ArchivedPage (*args, **kwargs)
    Bases: bigbuild.models.base.BasePage

    An archived custom page.

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    delete ()
        Delete the page directory.

    frontmatter_path
        Returns the metadata.md path where this page will be configured.

class bigbuild.models.PageList
    Bases: _abcoll.Sequence

    A list of all the Page and ArchivedPage objects in the application.

    get_archived_pages ()
        Returns a list of ArchivedPage objects ready to be built in this environment.

    get_directory_list (path)
        Returns the list of slugged page modules in the provided directory.

    get_dynamic_pages ()
        Returns a list of Page objects ready to be built in this environment.

    static get_page (slug, pagetype)
        Returns a list of Page objects from the provided slug directory.
```


b

`bigbuild.context_processors`, 1
`bigbuild.exceptions`, 1
`bigbuild.models`, 2

A

ArchivedPage (class in *bigbuild.models*), 2
ArchivedPage.DoesNotExist, 3
ArchivedPage.MultipleObjectsReturned, 3

B

BadMetadata (class in *bigbuild.exceptions*), 1
bigbuild() (in module *bigbuild.context_processors*),
1
bigbuild.context_processors (module), 1
bigbuild.exceptions (module), 1
bigbuild.models (module), 2

C

create_directory() (*bigbuild.models.Page*
method), 2

D

delete() (*bigbuild.models.ArchivedPage* method), 3
delete() (*bigbuild.models.Page* method), 2

F

frontmatter_path (*bigbuild.models.ArchivedPage*
attribute), 3
frontmatter_path (*bigbuild.models.Page* attribute),
2

G

get_archived_pages() (*bigbuild.models.PageList*
method), 3
get_context_data() (*big-*
build.exceptions.MissingRecommendedMetadataWarning
method), 1
get_directory_list() (*bigbuild.models.PageList*
method), 3
get_dynamic_pages() (*bigbuild.models.PageList*
method), 3
get_page() (*bigbuild.models.PageList* static method),
3

get_template() (*bigbuild.models.Page* method), 2

M

MissingRecommendedMetadataWarning (class
in *bigbuild.exceptions*), 1

P

Page (class in *bigbuild.models*), 2
Page.DoesNotExist, 2
Page.MultipleObjectsReturned, 2
PageList (class in *bigbuild.models*), 3

W

write_checklist() (*bigbuild.models.Page*
method), 2
write_frontmatter() (*bigbuild.models.Page*
method), 2
write_index() (*bigbuild.models.Page* method), 2
write_static() (*bigbuild.models.Page* method), 2