

Django Ease

Release 1.0.0

Satheesh Kumar

Oct 30, 2020

1 Overview **3**

1.1 Supported versions. 3

1.2 Package installation. 3

2 Modules **5**

2.1 Models. 5

Python Module Index **7**

Index **9**

Django Ease is a python package that contains most commonly used modules in django development. Instead of rewriting the commonly used modules the developers can import them from decore package.

Overview

The `decore` package should be installed properly in the current python environment to import and use it inside a Django app.

1.1 Supported versions

The following python versions are currently supported by this package

- 3.6
- 3.7
- 3.8

The below are the Django versions that are currently supported by this package

- 2.2
- 3.3

1.2 Package installation

`decore` package can be installed from the PyPi server directly into any of the supported python environments using,

```
$ pip install django-easy
```

Once the above command is completed you can check if the package is installed properly using,

```
$ python -m "import decore"
```

the above command should give the below exception in your python shell

```
django.core.exceptions.ImproperlyConfigured: Requested setting INSTALLED_APPS, but settings are not configured. You must either define the environment variable DJANGO_SETTINGS_MODULE or call settings.configure() before accessing settings.
```

if you get `ModuleNotFoundError`, this means that `django_easy` is not installed properly in your environment.

Modules

In this section we will know about all the modules that are available in decore package

2.1 Models

These are the development ready abstract models that are available inside decore package.

class `decore.models.Hidden (*args, **kwargs)`

Abstract model that can be inherited to have hide functionality in a model using `is_hidden` field.
Use `.hide()` to hide and `.unhide()` to unhide an object.

hide ()

hides the object if its not hidden

unhide ()

unhide the objects if its hidden

class `decore.models.Slug (*args, **kwargs)`

Abstract model that can be inherited to have a slug field in the model.

When inherited will check for `slug_from` attribute in the model, by default `SLUG_FROM` will have value `title`.

class `decore.models.TimeStamps (*args, **kwargs)`

Abstract model that can be inherited to track when an object was created or it was last updated.

`created_at` stores the datetime in which the object was created

`updated_at` stores the datetime in which the object was last updated

Both these fields are auto updated with their values on object creation and updation.

class `decore.models.UniqueSlug (*args, **kwargs)`

Abstract model that can be inherited to have a unique slug field. The slug will be generated from the field name you provide in `SLUG_FROM` attribute.

d

decore

[decore.models](#), 5

D

`decore.models`
module, [5](#)

H

`Hidden` (class in `decore.models`), [5](#)
`hide()` (`decore.models.Hidden` method), [5](#)

M

module
`decore.models`, [5](#)

S

`Slug` (class in `decore.models`), [5](#)

T

`TimeStamps` (class in `decore.models`), [5](#)

U

`unhide()` (`decore.models.Hidden` method), [5](#)
`UniqueSlug` (class in `decore.models`), [5](#)

