
django-observer Documentation

Release

Alisue

Sep 16, 2017

Contents

1	observer package	3
1.1	Subpackages	3
1.1.1	observer.tests package	3
1.1.1.1	Subpackages	3
1.1.1.2	Submodules	4
1.1.1.3	observer.tests.compat module	4
1.1.1.4	observer.tests.factories module	4
1.1.1.5	observer.tests.models module	4
1.1.1.6	observer.tests.test_investigator module	4
1.1.1.7	Module contents	4
1.1.2	observer.utils package	4
1.1.2.1	Submodules	4
1.1.2.2	observer.utils.models module	4
1.1.2.3	observer.utils.signals module	4
1.1.2.4	Module contents	5
1.1.3	observer.watchers package	5
1.1.3.1	Submodules	5
1.1.3.2	observer.watchers.auto module	5
1.1.3.3	observer.watchers.base module	5
1.1.3.4	observer.watchers.related module	5
1.1.3.5	observer.watchers.value module	5
1.1.3.6	Module contents	5
1.2	Submodules	5
1.3	observer.compat module	5
1.4	observer.conf module	5
1.5	observer.decorators module	5
1.6	observer.investigator module	5
1.7	observer.models module	6
1.8	observer.shortcuts module	6
1.9	Module contents	6
2	Indices and tables	7
	Python Module Index	9

Contents:

observer package

Subpackages

observer.tests package

Subpackages

observer.tests.test_utils package

Submodules

observer.tests.test_utils.test_models module

observer.tests.test_utils.test_signals module

Module contents

observer.tests.test_watchers package

Submodules

observer.tests.test_watchers.test_base module

observer.tests.test_watchers.test_related module

observer.tests.test_watchers.test_value module

Module contents

Submodules

`observer.tests.compat` module

`observer.tests.factories` module

`observer.tests.models` module

`observer.tests.test_investigator` module

Module contents

`observer.utils` package

Submodules

`observer.utils.models` module

`observer.utils.signals` module

`observer.utils.signals.register_reciever` (*model*, *signal*, *receiver*, *sender=None*, ***kwargs*)

Connect signal and receiver of the model without duplication

Parameters

- **model** (*class*) – A target class
- **signal** (*signal*) – A django signal
- **receiver** (*reciever*) – A django signal receiver
- **sender** (*model*) – A model class
- ****kwargs** – Options passed to the `signal.connect` method

Returns True for new registration, False for already registered.

Return type bool

`observer.utils.signals.unregister_reciever` (*model*, *signal*, *receiver*)

Disconnect signal and receiver of the model without exception

Parameters

- **model** (*class*) – A target class
- **signal** (*signal*) – A django signal
- **receiver** (*reciever*) – A django signal receiver

Returns True for success, False for already disconnected.

Return type bool

Module contents

observer.watchers package

Submodules

observer.watchers.auto module

observer.watchers.base module

observer.watchers.related module

observer.watchers.value module

Module contents

Submodules

observer.compat module

observer.compat.**lru_cache** (*maxsize*)

observer.conf module

observer.decorators module

observer.decorators.**watch** (*attr, callback, **kwargs*)

A decorator function for watching model attribute

observer.investigator module

class observer.investigator.**Investigator** (*model, include=None, exclude=None*)

Bases: object

A model modification investigator

Create an instance of investigator with a model class and call ‘prepare’ method just before save the model. After the model is saved, call ‘investigate’ method and the method will yields the field names modified.

get_cached (*pk, ignore_exception=True*)

Get cached object

Parameters

- **pk** (*any*) – A primary key of the object
- **ignore_exception** (*bool*) – Return None if the object is not found, if this is False, `ObjectDoesNotExist` raised

Raises `ObjectDoesNotExist` – When a specified object does not exists in the cache and *ignore_exception* is False.

Returns object or None

get_object (*pk*, *ignore_exception=True*)

Get latest object

It try to get the latest (unsaved) object from the database. If *ignore_exception* is True, return None, otherwise it raise `ObjectDoesNotExist` exception when no object is found.

Parameters

- **pk** (*any*) – A primary key of the object
- **ignore_exception** (*bool*) – Return None if the object is not found. if this is False, `ObjectDoesNotExist` raised.

Raises `ObjectDoesNotExist` – When a specified object does not exists *ignore_exception* is False.

Returns object or None

investigate (*instance*)

Call this function after the model instance is saved. It yield a name of modified attributes

prepare (*instance*)

Call this function before save the model instance

observer.models module

observer.shortcuts module

`observer.shortcuts.watch` (*model*, *attr*, *callback*, ***kwargs*)

A shortcut function for watching model attribute

Module contents

CHAPTER 2

Indices and tables

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

0

- `observer`, 6
- `observer.compat`, 5
- `observer.decorators`, 5
- `observer.investigator`, 5
- `observer.models`, 6
- `observer.shortcuts`, 6
- `observer.utils`, 5
- `observer.utils.signals`, 4
- `observer.watchers`, 5

G

`get_cached()` (`observer.investigator.Investigator` method),
5

`get_object()` (`observer.investigator.Investigator` method),
6

I

`investigate()` (`observer.investigator.Investigator` method),
6

`Investigator` (class in `observer.investigator`), 5

L

`lru_cache()` (in module `observer.compat`), 5

O

`observer` (module), 6

`observer.compat` (module), 5

`observer.decorators` (module), 5

`observer.investigator` (module), 5

`observer.models` (module), 6

`observer.shortcuts` (module), 6

`observer.utils` (module), 5

`observer.utils.signals` (module), 4

`observer.watchers` (module), 5

P

`prepare()` (`observer.investigator.Investigator` method), 6

R

`register_reciever()` (in module `observer.utils.signals`), 4

U

`unregister_reciever()` (in module `observer.utils.signals`), 4

W

`watch()` (in module `observer.decorators`), 5

`watch()` (in module `observer.shortcuts`), 6