thecut-authorship Documentation

Release 1.1

The Cut Creative

Contents

1	Welc 1.1 1.2 1.3 1.4	ome to thecut-authorship 3 Features 3 Documentation 3 Quickstart 3 Credits 4
2	Insta	llation instructions 5
3	Usag 3.1 3.2 3.3 3.4 3.5	Creating a model
4	A DT 1	Reference 9
	4.1 4.2 4.3 4.4 4.5 4.6 4.7	Models 9 View mixins 9 API view mixins 9 ModelForm mixins 10 Django admin mixins 10 Utilities 10 Settings 10
5	Testi	ng 11
	5.1 5.2	Running unit tests
6	Histo	ry 15
	6.1 6.2 6.3 6.4 6.5 6.6 6.7	1.2 (2017-05-09) 15 1.1 (2016-12-21) 15 1.0 (2016-08-16) 15 0.11 (2016-08-16) 15 0.10.1 (2015-11-19) 16 0.10 (2015-10-09) 16 0.9 (2015-08-25) 16 0.7.1 (2014-12-11) 16

Рy	thon I	Module Index																		21
7	Cred	its																		19
	6.13	0.5 (2013-03-15)		 							•									17
	6.12	0.5.1 (2014-03-19) .	 								 								17
	6.11	0.5.2 (2014-06-20) .	 								 								17
	6.10	0.5.3 (2014-07-09) .	 								 								16
	6.9	0.7 (2014-11-24)		 								 								16

Contents:

Contents 1

2 Contents

Welcome to thecut-authorship

A set of Django mixins to easily record authorship information for your models.

Features

- Base model allows easy recording of authorship information.
- Integration with Django's class-based views and forms.
- Integration with Django's admin.

Documentation

The full documentation is at https://thecut-authorship.readthedocs.org.

Quickstart

Install the cut-authorship using the installation instructions found in the project documentation.

Build a model based on the cut. authorship. models. Authorship to record authorship information on it:

```
from thecut.authorship.models import Authorship

class MyModel(Authorship):
    pass
```

This adds created_by, created_at, updated_by, and updated_at to your model.

Pass a user into calls to .save() to record which user changed the object:

```
example = MyModel()
example.save(user=request.user)
```

If you need to update model data and there's no direct link to a website user, generate and use a site-wide 'generic' user.:

```
from thecut.authorship.models import get_website_user
example = MyModel()
example.save(user=get_website_user())
```

If you wish to automatically record authorship information for changes made in the Django admin, use thecut. authorship.admin.AuthorshipMixin.:

```
from .models import MyModel
from django.contrib import admin
from thecut.authorship.admin import AuthorshipMixin

@admin.register(MyModel)
class MyModelAdmin(AuthorshipMixin, admin.ModelAdmin):
    pass
```

If you wish to integrate with django.forms.ModelForm, use thecut.authorship.forms. AuthorshipMixin and thecut.authorship.views.AuthorshipMixin.

In your forms.py:

```
from .models import MyModel
from django import forms
from thecut.authorship.forms import AuthorshipMixin

class MyModelForm(forms.ModelForm):

    class Meta(object):
        model = MyModel
```

In your views.py:

```
from .forms import MyModelForm
from .models import MyModel
from django.views.generic import CreateView
from thecut.authorship.views import AuthorshipMixin

class MyModelCreateView(AuthorshipMixin, CreateView):
    form_class = MyModelForm
```

MyModelCreateView will now automatically pass request.user through to MyModelForm, which will pass it through to the model's <code>save()</code> method.

Credits

See AUTHORS.rst.

Installation instructions

1. Install via pip / pypi:

```
$ pip install thecut-authorship
```

2. Add to your project's INSTALLED_APPS setting:

```
INSTALLED_APPS = [
    # ...
    'thecut.authorship'
    # ...
]
```

3. Sync your project's migrations:

```
$ python manage.py migrate authorship
```

thecut-authorship Documentation, Release 1.1	

Usage

Creating a model

Subclass the cut.authorship.models.Authorship to create the necessary fields:

```
class MyModel(Authorship):
    pass
```

Saving authorship information

the cut.authorship.models.Authorship.save() will populate the authorship fields when necessary:

```
m = MyCoolModel()
m.save(user=request.user)
```

This will update the model's thecut.authorship.models.Authorship.updated_by and thecut.authorship.models.Authorship.updated_at fields automatically, as well as thecut.authorship.models.Authorship.created_by and thecut.authorship.models.Authorship.created_at if the object is new.

If the model creation / update doesn't directly relate to a user, use the site-wide generic authorship user. This can be retrieved with thecut.authorship.utils.get_website_user():

```
m = MyCoolModel()
m.save(user=get_website_user())
```

Tip: If you specify update_fields in your call to thecut.authorship.models.Authorship. save(), the list will automatically be updated to ensure that authorship information is saved.

Integrating with django.contrib.admin

You can automatically update authorship information when a model is altered in the Django admin interface by using thecut.authorship.admin.AuthorshipMixin:

```
class MyAdmin(AuthorshipMixin, admin.ModelAdmin):
    pass
```

Hint: This also applies to child inlines if they refer to subclasses of thecut.authorship.models. Authorship.

Integrating with class-based views and ModelForms

Use the cut. authorship. views. Authorship Mixin on your django. views. generic.edit. Model Form Mixin-based views (django. views. generic.edit. Create View, django. views. generic.edit. Update View. etc):

```
class MyModelCreateView(AuthorshipMixin, CreateView):
    form_class = MyModelForm
```

Then, use the cut.authorship.forms.AuthorshipMixin on your django.forms. ModelForm-based forms:

```
class MyModelForm(AuthorshipMixin, ModelForm):
    class Meta(object):
        model = MyModel
```

Together, these mixins will—upon a successful form submission—appropriately record request.user on the target object.

Warning: You must use the cut.authorship.views.AuthorshipMixin on the view and the cut.authorship.forms.AuthorshipMixin on the form for this to work.

Integrating with Django REST Framework

Use thecut.authorship.api.views.AuthorshipMixin on your CreateModelMixin / UpdateModelMixin-based API views.

8 Chapter 3. Usage

API Reference

Models

```
class thecut.authorship.models.Authorship(*args, **kwargs)
   Abstract model to track when an instance was created/updated and by whom.

created_at = None
    datetime for when this object was first created.

created_by
    User who first created this object (required).

save (user=None, **kwargs)
    A custom Model.save() method that appropriately populates authorship fields.

updated_at = None
    datetime for when this object was last saved.

updated_by
    User who last saved this object (required).
```

View mixins

class thecut.authorship.views.AuthorshipMixin
 Adds the request's User instance to the form kwargs.

API view mixins

class the cut.authorship.api.views.AuthorshipMixin

ModelForm mixins

class thecut.authorship.forms.AuthorshipMixin(user, *args, **kwargs)

Mixin for a ModelForm which sets created_by and updated_by fields for the instance when saved.

Requires that a User instance be passed in to the constructor. Views which utilise AuthorshipViewMixin handle this already.

Django admin mixins

class thecut.authorship.admin.AuthorshipMixin

Mixin for a model admin to set created/updated by on save.

Utilities

class thecut.authorship.utils.get_website_user

Get a generic 'website' user.

Can be used to specify the required user when there is no direct link to a real user.

Settings

thecut.authorship.settings.AUTH USER MODEL = 'auth.User'

The user model that thecut.authorship.utils.get_website_user() will query against. Defaults to settings.AUTH_USER_MODEL.

thecut.authorship.settings.WEBSITE_USER = {u'username': u'website'}

A dictionary that thecut.authorship.utils.get_website_user() will pass to get or create() in order to return the generic website user.

Testing

Running unit tests

Using your system's Python / Django

You can perform basic testing against your system's Python / Django.

1. Install the test suite requirements:

```
$ pip install -r requirements-test.txt
```

2. Ensure a version of Django is installed:

```
$ pip install Django
```

3. Run the test runner:

```
$ python runtests.py
```

Using a virtualenv

You can use virtualenv to test without polluting your system's Python environment.

1. Install virtualenv:

```
$ pip install virtualenv
```

2. Create and activate a virtualenv:

```
$ cd thecut-authorship
$ virtualenv .
$ source bin/activate
(thecut-authorship) $
```

3. Follow 'Using your system's Python / Django' above.

Using tox

You can use tox to automatically test the application on a number of different Python and Django versions.

1. Install tox:

```
$ pip install -r requirements-test.txt
```

2. Run tox:

```
(thecut-authorship) $ tox --recreate
```

Tox assumes that a number of different Python versions are available on your system. If you do not have all required versions of Python installed on your system, running the tests will fail. See tox.ini for a list of Python versions that are used during testing.

Test coverage

The included tox configuration automatically detects test code coverage with coverage:

```
$ coverage report
```

Available tests

Forms

class thecut.authorship.tests.test_forms.TestAuthorshipMixin (methodName='runTest')

test_requires_an_extra_argument_on_creating_an_instance()

Ensure that the cut.authorship.forms.AuthorshipMixin-based forms cannot be instantiated without passing in a user.

test_sets_user_attribute()

Ensure that the cut.authorship.forms.AuthorshipMixin-based forms properly set the cut.authorship.forms.AuthorshipMixin.user when one is passed on instantiation.

class thecut.authorship.tests.test_forms.TestAuthorshipMixinSave (methodName='runTest')

```
test_calls_super_class_save_method(*args, **keywargs)
```

Ensure that the cut.authorship.forms.AuthorshipMixin.save() calls the superclass's save method..

```
test_does_not_set_created_by_if_instance_is_saved(*args, **keywargs)
```

Ensure that the cut. authorship. forms. Authorship Mixin-based forms do not set the cut. authorship. models. Authorship Mixin. created_by if the target object has already been saved.

12 Chapter 5. Testing

test_sets_created_by_if_instance_is_not_saved(*args, **keywargs)

Ensure that thecut.authorship.forms.AuthorshipMixin-based forms appropriately set thecut.authorship.models.AuthorshipMixin.created_by when a user is provided and the target object has not been saved before.

test_sets_updated_by_to_given_user(*args, **keywargs)

Ensure that thecut.authorship.forms.AuthorshipMixin-based forms appropriately set thecut.authorship.models.AuthorshipMixin.updated_by when a user is provided.

Models

class thecut.authorship.tests.test_models.TestAuthorshipModel (methodName='runTest')

test_does_not_change_created_by_when_model_instance_is_saved()

Ensure that the cut.authorship.models.Authorship.created_by is not updated for existing models.

test_sets_created_by_when_model_instance_is_first_saved()

Check if created_by is correctly set on first save.

test_sets_updated_at_if_update_fields_is_specified()

Ensure that the cut.authorship.models.Authorship.updated_at is updated, even when update_fields is specified.

test_sets_updated_by_if_update_fields_is_specified()

Ensure that the cut.authorship.models.Authorship.updated_by is updated, even when update_fields is specified.

test_sets_updated_by_when_model_instance_is_saved()

Ensure that the cut.authorship.models.Authorship.updated_by is updated on save.

Utils

class thecut.authorship.tests.test_utils.TestGetWebsiteUser (methodName='runTest')
 Tests for get_website_user().

test_get_website_user_returns_same_user()

Test if the same user is returned over multiple calls.

test_get_website_user_returns_user()

Test if something is returned.

5.2. Available tests

14 Chapter 5. Testing

History

1.2 (2017-05-09)

- Added support for Django 1.11 and Python 3.6.
- Drop support for Django 1.9.

1.1 (2016-12-21)

• Added support for Django REST Framework.

1.0 (2016-08-16)

- Removed deprecated APIs.
- Removed compatibility code for unsupported versions of Django.
- Improved test coverage.
- Removed code paths in Authorship.save() that could not logically be reached.

0.11 (2016-08-16)

- Rewrote documentation.
- Redesigned testing environment.

0.10.1 (2015-11-19)

- Fixed bug when saving on Django 1.4
- Started using unittest from Python standard library. Removes Python < 2.7 support.
- Updated tox configuration to test against newer versions of Django / Python.
- Fixed bug that stopped authorship information being updated when update_fields is defined but empty.

0.10 (2015-10-09)

- Test against Django 1.8
- Fixed bug where models were incorrectly detected as 'not new' (for the purpose of setting created_at and created_by) when a pk is manually specified.

0.9 (2015-08-25)

• Set "on_delete=models.PROTECT" on authorship fields that relate to users.

0.7.1 (2014-12-11)

- Ensure that created_at and created_by are updated regardless of the contents of update_fields.
- · Added Django admin mixin to save authorship information when using inlines.

0.7 (2014-11-24)

- Updated documentation.
- Removed Makefile.
- Altered testing environment to support Django 1.7
- Added Django 1.7 AppConfig.

0.5.3 (2014-07-09)

- Added unit tests for model and form mixin.
- Improved Python 3 compatibility.
- Updated test environment to test against newer versions of Django.
- Ensure that updated_at and updated_by are updated regardless of the contents of update_fields.

16 Chapter 6. History

0.5.2 (2014-06-20)

• Added AuthorshipFactory for testing.

0.5.1 (2014-03-19)

• Removed distribute from the install_requires list.

0.5 (2013-03-15)

• First release.

18 Chapter 6. History

$\mathsf{CHAPTER}\ 7$

Credits

- Josh Crompton <josh.crompton@thecut.net.au>
- Elena Williams <elena.williams@thecut.net.au>
- Matt Austin <matt.austin@thecut.net.au>
- Kye Russell <kye.russell@thecut.net.au>
- Mark Lockett <mark.lockett@thecut.net.au>

20 Chapter 7. Credits

Python Module Index

```
thecut.authorship.settings, 10
```

thecut-authorship De	ocumentation,	Release	1.1
----------------------	---------------	---------	-----

22 Python Module Index

Index

AUTH_USER_MODEL (in module the cut.authorship.settings), 10 Authorship (class in thecut.authorship.models), 9 AuthorshipMixin (class in thecut.authorship.admin), 10 AuthorshipMixin (class in thecut.authorship.api.views), 9 AuthorshipMixin (class in thecut.authorship.orms), 10 AuthorshipMixin (class in thecut.authorship.views), 9 CC created_at (thecut.authorship.models.Authorship attribute), 9 created_by (thecut.authorship.models.Authorship tribute), 9 Created_by (thecut.authorship.models.Authorship tribute), 9 Created_by (thecut.authorship.models.Authorship attribute), 9 Created_by (thecut.authorship.models.Authorship method), 13 test_sets_updated_by_if_update_fields_is_specified() (thecut.authorship.tests.test_models.TestAuthorshipModel method), 13 test_sets_updated_by_to_given_user() (thecut.authorship.tests.test_forms.TestAuthorshipMixin method), 13 test_sets_updated_by_when_model_instance_is_saved() (thecut.authorship.tests.test_forms.TestAuthorshipModel method), 13 test_sets_updated_by_when_model_instance_is_saved() (thecut.authorship.tests.test_forms.TestAuthorshipMixin method), 13 test		
Authorship (class in thecut.authorship.models), 9 AuthorshipMixin (class in thecut.authorship.admin), 10 AuthorshipMixin (class in thecut.authorship.adpi.views), 9 AuthorshipMixin (class in thecut.authorship.views), 9 C created_at (thecut.authorship.models.Authorship atribute), 9 created_by (thecut.authorship.models.Authorship atribute), 9 Created_by (thecut.authorship.models.Authorship models.Authorship method), 13 test_sets_updated_by_if_update_fields_is_specified() (thecut.authorship.tests.test_models.TestAuthorshipModel method), 13 test_sets_updated_by_if_update_fields_is_specified() (thecut.authorship.tests.test_models.TestAuthorshipModel method), 13 test_sets_updated_by_if_update_fields_is_specified() (thecut.authorship.tests.test_models.TestAuthorshipModel method), 13 test_sets_updated_by_if_update_fields_is_specified() (thecut.authorship.tests.test_models.TestAuthorshipModel method), 13 test_sets_updated_by_to_given_user() (thecut.authorship.tests.test_forms.TestAuthorshipModel method), 13 test_sets_updated_by_to_given_user() (thecut.authorship.tests.test_models.TestAuthorshipModel method), 13 test_sets_updated_by_to_given_user() (thecut.authorship.tests.test_forms.TestAuthorshipModel method), 13 test_sets_updated_by_to_given_user() (thecut.authorship.tests.test_forms.TestA	out outhorship softings) 10	
AuthorshipMixin (class in thecut.authorship.admin), 10 AuthorshipMixin (class in thecut.authorship.forms), 10 AuthorshipMixin (class in thecut.authorship.forms), 10 AuthorshipMixin (class in thecut.authorship.forms), 10 AuthorshipMixin (class in thecut.authorship.views), 9 CC created_at (thecut.authorship.models.Authorship tribute), 9 created_by (thecut.authorship.models.Authorship method), 13 Cs (thecut.authorship.tests.test_models.TestAuthorshipModel method), 13 Cs (thecut.authorship.tests.test_forms.TestAuthorshipModel method), 13 Cs (thecut.authorship.tests.test_forms.TestAuthorshipModel method), 13 cst_sets_updated_by_if_update_fields_is_specified() (thecut.authorship.tests.test_models.TestAuthorshipModel method), 13 cst_sets_updated_by_if_update_fields_is_specified() (thecut.authorship.tests.test_models.TestAuthorshipModel method), 13 cst_sets_updated_by_if_update_fields_is_specified() (thecut.authorship.tests.test_forms.TestAuthorshipModel method), 13 cst_sets_updated_by_if_update_fields_is_specified() (thecut.authorship.tests.test_forms.TestAuthorshipModel method), 13 cst_sets_updated_by_if_update_fields_is_specified() (thecut.authorship.tests.test_forms.TestAuthorshipModel method), 13 cst_sets_updated_by_if_update_fields_is_is_specified() (thecut.authorship.tests.test_forms.TestAuthorshipModel method), 13 cst_sets_updated_by_if_update_fields_is_is_apecified() (thecut.authorship.tests.test_forms.TestAuthorshipModel method), 13 cst_sets_updated_by_if_update_fields_is_is_apecified() (thecut.authorship.tests.test_forms.TestAuthorshipModel method), 13 cst_sets_updated_by_if_update_fields_is_is_apecified() (thecut.authorship.tests.test_forms.TestAuthorshipModel method), 13 cst_sets_updated_by_if_update_fields_is_i		test_sets_created_by_if_instance_is_not_saved() (the-
AuthorshipMixin (class in thecut.authorship.forms), 10 AuthorshipMixin (class in thecut.authorship.views), 9 C created_at (thecut.authorship.models.Authorship tribute), 9 created_by (thecut.authorship.models.Authorship tribute), 9 C attribute), 9 Created_by (thecut.authorship.models.Authorship tribute), 9 Created_by (thecut.authorship.models.Authorship tribute), 9 C attribute), 13 C attribute), 14 C attribute), 14 C attribute), 13 C attribute), 14 C attribute), 14 C attribute), 13 C attribute), 14 C attribute), 15 C attri	AuthorshipMixin (class in thecut.authorship.admin), 10	
AuthorshipMixin (class in thecut.authorship.views), 9 C created_at (thecut.authorship.models.Authorship tribute), 9 Created_by (thecut.authorship.models.Authorship tribute), 9 C created_by (thecut.authorship.models.Authorship attribute), 9 C created_by (thecut.authorship.models.Authorship method), 13 E created_by (thecut.authorship.models.Authorship.utils), 10 S created_by (thecut.authorship.models.Authorship.utils), 10 E created_by (thecut.authorship.models.Authorship.utils), 10 C created_by (thecut.authorship.tests.test_forms.TestAuthorshipModel method), 13 E cst_sets_updated_by_tif_update_fields_is_specified() (thecut.authorship.tests.test_models.TestAuthorshipModel method), 13 E cst_sets_updated_by_to_given_user() (the-cut.authorship.tests.test_forms.TestAuthorshipModel method), 13 E cst_sets_updated_by_tiests.test_models.TestAuthorshipModel (thecut.authorship.tests.test_forms.TestAuthorshipModel (thecut.authorship.tests.test_forms.TestAuthorshipModel (class in thecut.authorship.tests.test_forms), 12 E cst_sets_updated_by_tie_update_fields_is_specified() (thecut.authorship.tests.test_models.TestAuthorshipModel (class in thecut.		
created_at (thecut.authorship.models.Authorship at tribute), 9 created_by (thecut.authorship.models.Authorship tribute), 9 created_by (thecut.authorship.models.Authorship at tribute), 9 G G G Set_website_user (class in thecut.authorship.utils), 10 S save() (thecut.authorship.models.Authorship method), 9 T T Set_calls_super_class_save_method() (thecut.authorship.tests.test_forms.TestAuthorshipMixinSave method), 12 test_does_not_change_created_by_when_model_instance_is_saved()	AuthorshipMixin (class in thecut.authorship.views), 9	(thecut.authorship.tests.test_models.TestAuthorshipModel method), 13
created_at (thecut.authorship.models.Authorship tribute), 9 created_by (thecut.authorship.models.Authorship tribute), 9 G get_website_user (class in thecut.authorship.utils), 10 S save() (thecut.authorship.models.Authorship method), 9 T test_calls_super_class_save_method() (thecut.authorship.tests.test_forms.TestAuthorshipMixin test_calls_super_class_save_method() (thecut.authorship.tests.test_forms.TestAuthorshipMixin (class in thecut.authorship.tests.test_forms.TestAuthorshipMixin (class in thecut.authorship.tests.test_forms), 12 test_does_not_change_created_by_when_model_instance_is_saved() (thecut.authorship.tests.test_forms), 12 method), 13 test_sets_updated_by_if_update_fields_is_specified() (thecut.authorship.tests.test_models.TestAuthorshipMixinSave method), 13 test_sets_updated_by_to_given_user() (thecut.authorshipMixinSave method), 13 test_sets_updated_by_when_model_instance_is_saved() (thecut.authorship.tests.test_models.TestAuthorshipModel cut.authorship.tests.test_forms.TestAuthorshipMixin method), 13 test_sets_updated_by_to_given_user() (thecut.authorshipMixinSave method), 13 test_sets_updated_by_to_given_user() (thecut.authorshipMixinSave) (thecut.authorshipMixinSave) (thecut.authorshipMixinSave) (thecut.authorshipMixinSave) (thecut.authorshipMixinSave) (thecut.authorshipMixinSave) (thecut.authorshipMixinSave) (thecut.authorshipMixinSave) (thecut.authorshipMixinSave) (C	
created_by (thecut.authorship.models.Authorship tribute), 9 G	tribute) 0	method), 13
method), 13 G test_sets_updated_by_to_given_user() (the-cut.authorship.utils), 10 S test_sets_updated_by_when_model_instance_is_saved() (thecut.authorship.tests.test_models.TestAuthorshipModel save() (thecut.authorship.models.Authorship method), 9 T test_calls_super_class_save_method() (the-cut.authorship.tests.test_forms.TestAuthorshipMixin test_calls_super_class_save_method() (the-cut.authorship.tests.test_forms.TestAuthorshipMixin test_does_not_change_created_by_when_model_instance_i3estAuthorshipMixinSave (class in the-(thecut.authorship.tests.test_models.TestAuthorshipModel cut.authorship.tests.test_forms), 12 method), 13 TestAuthorshipModel (class in the-test_does_not_set_created_by_if_instance_is_saved() cut.authorship.tests.test_models), 13		
G test_sets_updated_by_to_given_user() (the-cut.authorship.utils), 10 method), 13 S test_sets_updated_by_when_model_instance_is_saved() (thecut.authorship.models.Authorship method), 9 method), 13 T test_sets_updated_by_when_model_instance_is_saved() (thecut.authorship.tests.test_models.TestAuthorshipModel method), 13 T test_sets_updated_by_when_model_instance_is_saved() (thecut.authorship.tests.test_models.TestAuthorshipModel method), 13 test_sets_user_attribute() (thecut.authorship.tests.test_forms.TestAuthorshipMixin method), 12 cut.authorship.tests.test_forms.TestAuthorshipMixinstantenthorshipMixinstantenthorship.tests.test_forms), 12 test_does_not_change_created_by_when_model_instance_is_saved() cut.authorship.tests.test_forms), 12 method), 13 TestAuthorshipModel (class in thetets_does_not_set_created_by_if_instance_is_saved() cut.authorship.tests.test_models), 13	tribute), 9	· · · · · · · · · · · · · · · · · · ·
cut.authorship.tests.test_forms.TestAuthorshipMixinSave method), 13 S test_sets_updated_by_when_model_instance_is_saved() (thecut.authorship.models.Authorship method), 9 method), 13 T test_sets_updated_by_when_model_instance_is_saved() (thecut.authorship.tests.test_models.TestAuthorshipModel method), 13 test_sets_user_attribute() cut.authorship.tests.test_forms.TestAuthorshipMixin method), 12 cut.authorship.tests.test_forms.TestAuthorshipMixInsAuthorshipMixin (class in themethod), 12 test_does_not_change_created_by_when_model_instance_is_saved() (thecut.authorship.tests.test_forms), 12 method), 13 TestAuthorshipModel cut.authorship.tests.test_forms), 12 method), 13 TestAuthorshipModel (class in themethod), 13	G t	test_sets_updated_by_to_given_user() (the-
(thecut.authorship.tests.test_models.TestAuthorshipModel save() (thecut.authorship.models.Authorship method), 9 method), 13 Test_sets_user_attribute() (the-cut.authorship.tests.test_forms.TestAuthorshipMixin method), 12 cut.authorship.tests.test_forms.TestAuthorshipMixInstAuthorshipMixin (class in themethod), 12 cut.authorship.tests.test_forms.TestAuthorshipMixin (class in themethod), 12 test_does_not_change_created_by_when_model_instance_is_setAuthorshipMixinSave (class in the-(thecut.authorship.tests.test_models.TestAuthorshipModel cut.authorship.tests.test_forms), 12 method), 13 TestAuthorshipModel (class in the-test_does_not_set_created_by_if_instance_is_saved() cut.authorship.tests.test_models), 13		<u>.</u>
(thecut.authorship.models.Authorship method), 9 method), 13 Total dest_sets_user_attribute() (the-cut.authorship.tests.test_forms.TestAuthorshipMixin method), 12 Cut.authorship.tests.test_forms.TestAuthorshipMixInstAvethorshipMixin (class in the-method), 12 Cut.authorship.tests.test_forms.TestAuthorshipMixInstAvethorshipMixin (class in the-method), 12 test_does_not_change_created_by_when_model_instance_iiTestAvethorshipMixinSave (class in the-(thecut.authorship.tests.test_models.TestAuthorshipModel cut.authorship.tests.test_forms), 12 method), 13 TestAuthorshipModel (class in the-test_does_not_set_created_by_if_instance_is_saved() cut.authorship.tests.test_models), 13	S	
test_sets_user_attribute() (the- cut.authorship.tests.test_forms.TestAuthorshipMixin test_calls_super_class_save_method() (the- cut.authorship.tests.test_forms.TestAuthorshipMixin (class in the- method), 12 cut.authorship.tests.test_forms.TestAuthorshipMixin (class in the- method), 12 test_does_not_change_created_by_when_model_instance_is_savet()orshipMixinSave (class in the- (thecut.authorship.tests.test_models.TestAuthorshipModel cut.authorship.tests.test_forms), 12 method), 13 TestAuthorshipModel (class in the- test_does_not_set_created_by_if_instance_is_saved() cut.authorship.tests.test_models), 13		<u>-</u>
cut.authorship.tests.test_forms.TestAuthorshipMixin test_calls_super_class_save_method() (the- cut.authorship.tests.test_forms.TestAuthorshipMixInstAvethorshipMixin (class in the- method), 12 test_does_not_change_created_by_when_model_instance_iiTestAvethorshipMixinSave (class in the- (thecut.authorship.tests.test_models.TestAuthorshipModel cut.authorship.tests.test_forms), 12 method), 13 TestAuthorshipModel (class in the- test_does_not_set_created_by_if_instance_is_saved() cut.authorship.tests.test_models), 13	t t	
cut.authorship.tests.test_forms.TestAuthorshipMixInstAvethorshipMixin (class in method), 12 cut.authorship.tests.test_forms), 12 test_does_not_change_created_by_when_model_instance_is_extAuthorshipMixinSave (class in the-(thecut.authorship.tests.test_models.TestAuthorshipModel cut.authorship.tests.test_forms), 12 method), 13 TestAuthorshipModel (class in the-test_does_not_set_created_by_if_instance_is_saved() cut.authorship.tests.test_models), 13	1	$cut. authorship. tests. test_forms. TestAuthorship Mixin\\$
method), 12 cut.authorship.tests.test_forms), 12 test_does_not_change_created_by_when_model_instance_is_estAuthorshipMixinSave (class in the- (thecut.authorship.tests.test_models.TestAuthorshipModel cut.authorship.tests.test_forms), 12 method), 13 TestAuthorshipModel (class in the- test_does_not_set_created_by_if_instance_is_saved() cut.authorship.tests.test_models), 13		
test_does_not_change_created_by_when_model_instance_is_estAuthorshipMixinSave (class in the- (thecut.authorship.tests.test_models.TestAuthorshipModel cut.authorship.tests.test_forms), 12 method), 13 TestAuthorshipModel (class in the- test_does_not_set_created_by_if_instance_is_saved() cut.authorship.tests.test_models), 13		
(thecut.authorship.tests.test_models.TestAuthorshipModel cut.authorship.tests.test_forms), 12 method), 13 TestAuthorshipModel (class in thetest_does_not_set_created_by_if_instance_is_saved() cut.authorship.tests.test_models), 13		<u> </u>
test_does_not_set_created_by_if_instance_is_saved() cut.authorship.tests.test_models), 13	$(the cut. authorship. tests. test_models. TestAuthorshi$	pModel cut.authorship.tests.test_forms), 12
(thecut authorship tests test, forms Test Authorship Mister Website User (class in the-	(thecut.authorship.tests.test_forms.TestAuthorship.	
method), 12 cut.authorship.tests.test_utils), 13	·	
test_get_website_user_returns_same_user() (the-thecut.authorship.settings (module), 10	<u>-</u> e -	
cut.authorship.tests.test_utils.TestGetWebsiteUser method), 13	method), 13	U
test get website user returns user() (the- undeted at (theout authorship models Authorship at	test_get_website_user_returns_user() (the-	updated_at (thecut.authorship.models.Authorship at-
	<u>-</u>	· · · · · · · · · · · · · · · · · · ·
cut.authorship.tests.test_utils.TestGetWebsiteUser tribute), 9	memouj, 15	updated_by (thecut.authorship.models.Authorship at- tribute), 9

W

 $\begin{tabular}{ll} WEBSITE_USER (in module the cut. authorship. settings), \\ 10 \end{tabular}$

24 Index