django-inviter Documentation

Release 0.2.4

Alen Mujezinovic

March 28, 2012

CONTENTS

django-inviter allows you to invite users to your Django application. Invited users are saved as inactive users in your database and activated upon registration.

CHAPTER

ONE

INSTALLATION

```
pip install django-inviter
```

CHAPTER

TWO

CONFIGURATION

Add `inviter` and `django.contrib.sites` to your `INSTALLED_APPS`

```
INSTALLED_APPS = (
    'django.contrib.sites',
    'inviter'
)
```

Include `inviter.urls` into your root `urls.py` file under the `inviter` namespace

```
urlpatterns = patterns('',
    url('^invites/', include('inviter.urls', namespace = 'inviter'))
```

CHAPTER

THREE

USAGE

To invite people make use of `inviter.utils.invite`

```
from inviter.utils import invite

invite("foo@example.com", request.user, current_time = datetime.now())
```

`inviter.utils.invite` also allows you to make use of a custom email sending function, say to send HTML emails

```
from inviter.utils import invite

def sendhtml(invitee, inviter, **kwargs):
    # Load templates, send the email here
    pass

invite("foo@example.com", request.user, sendfn=sendhtml)
```

A useful application of this is keeping track of who invites whom:

```
from inviter import utils
from app.models import Invites

def send(invitee, inviter, **kwargs):
    Invites.objects.get_or_create(invitee = invitee, inviter = inviter)
    utils.send_invite(invitee, inviter, **kwargs)

utils.invite("foo@example.com", request.user, sendfn=send)
```

Consult `inviter.utils.invite` and `inviter.utils.send_invite` for more information.

By default `inviter.utils.send_invite` will render `inviter/email/subject.txt` and `inviter/email/body.txt` for the email.

`/inviter/register.html` and `inviter/done.html` are rendered when registering respectively when done.

If you need a post registration hook, override the registration form with the settings below.

CHAPTER

FOUR

SETTINGS

There are a couple of editable settings

INVITER_FORM

Default `inviter.forms.RegistrationForm`

Type str

The form to be used when an invited user signs up.

INVITER_REDIRECT

Default `'inviter:done'`

Type str

The URL to redirect the user to when the signup completes. This is either a URL to reverse via `reverse(INVITER_REDIRECT)` or a simple string. Reversing the URL is tried before using the string.

INVITER_TOKEN_GENERATOR

Default `'inviter.tokens.generator'`

Type str

The generator used to create a token which is used to assemble an invite URL

INVITER_FROM_EMAIL

Default `settings.DEFAULT_FROM_EMAIL`

The email address used to send invites from

CHAPTER

FIVE

API

5.1 Forms

class inviter.forms.**OptOutForm**(*data=None*, *files=None*, *auto_id='id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.util.ErrorList'>*, *label_suffix=':'*, *empty_permitted=False*, *instance=None*)

Dummy form for opting out.

save()

Delete the user object from the database and store the SHA1 hashed email address in the database to make sure this person does not receive any further invitation emails.

class inviter.forms.**RegistrationForm**(**args*, ***kwargs*)

The standard form that is displayed to users when registering. It gives a user the option to change their email address.

5.2 Models

class inviter.models.**OptOut**(**args*, ***kwargs*)

Opt-out email addresses are stored as SHA1 hashes to make sure we don't accidentally collect any more data once a person signalled they're not interested in receiving any more invitation emails from us.

5.3 URLs

inviter.urls.**urlpatterns = [<RegexURLPattern done ^done/$>, <RegexURLPattern register ^register/(?P<uidb36>[0-**

^done/$

Name done

^register/(?P<uidb36>[0-9A-Za-z]{1,13})-(?P<token>[0-9A-Za-z]+)/$

Name register

^optout/(?P<uidb36>[0-9A-Za-z]{1,13})-(?P<token>[0-9A-Za-z]+)/$

Name opt-out

^optout/done/$

Name opt-out-done

5.4 Utils

inviter.utils.**invite**(*email*, *inviter*, *sendfn=<function send_invite at 0x3ccbc08>*, *resend=True*, ***kwargs*)

Invite a given email address and return a `(User, sent)` tuple similar to the Django `django.db.models.Manager.get_or_create()` method.

If a user with `email` address does not exist:

- Creates a `django.contrib.auth.models.User` object
- Set `user.email = email`
- Set `user.is_active = False`
- Set a random password
- Send the invitation email
- Return `(user, True)`

If a user with `email` address exists and `user.is_active == False`:

- Re-send the invitation
- Return `(user, True)`

If a user with `email` address exists:

- Don't send the invitation
- Return `(user, False)`

If the email address is blocked:

- Don't send the invitation
- Return `(None, False)`

To customize sending, pass in a new `sendfn` function as documented by `inviter.utils.send_invite`:

```
sendfn = lambda invitee, inviter, **kwargs: 1
invite("foo@bar.com", request.user, sendfn = sendfn)
```

Parameters

- **email** – The email address
- **inviter** – The user inviting the email address
- **sendfn** – An email sending function. Defaults to `inviter.utils.send_invite`
- **resend** – Resend email to users that are not registered yet

inviter.utils.**send_invite**(*invitee*, *inviter*, *url=None*, *opt_out_url=None*, ***kwargs*)

Send the default invitation email assembled from `inviter/email/subject.txt` and `inviter/email/body.txt`

Both templates will receive all the `kwargs`.

Parameters

- **invitee** – The invited user
- **inviter** – The inviting user
- **url** – The invite URL

- **subject_template** – The template to render for the subject
- **body_template** – The template to render for the body
- **opt_out_url** – A URL where users can permanently opt out of invitations

5.5 Views

class inviter.views.**OptOut**(***kwargs*)
We want to give the user also the option to *not* receive any invitations anymore, which is happening in this view and inviter.forms.OptOutForm.

class inviter.views.**Register**(***kwargs*)
A registration view for invited users. The user model already exists - this view just takes care of setting a password and username, and maybe update the email address. Anywho - one can customize the form that is used.

form
alias of RegistrationForm

get(*request*, *user*)
Unfortunately just a copy of django.contrib.auth.views.password_reset_confirm

post(*request*, *user*)
Unfortunately just a copy of django.contrib.auth.views.password_reset_confirm

class inviter.views.**UserMixin**
Handles retrieval of users from the token and does a bit of access management.

dispatch(*request*, *uidb36*, *token*, **args*, ***kwargs*)
Overriding the default dispatch method on Django's views to do some token validation and if necessary deny access to the resource.

Also passes the user as first argument after the request argument to the handler method.

inviter.views.**import_attribute**(*path*)
Import an attribute from a module.

Made by Caffeinehit Ltd.

PYTHON MODULE INDEX

i