

---

# **django-inviter2 Documentation**

*Release 1.3.0*

**Alen Mujezinovic and Michael J Schultz**

May 04, 2016



<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Configuration</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>Settings</b>	<b>9</b>
<b>5</b>	<b>API</b>	<b>11</b>



django-inviter2 allows you to invite users to your Django application. Invited users are saved as inactive users in your database and activated upon registration.



---

## Installation

---

```
pip install django-inviter2
```





---

## Configuration

---

Add `inviter2` to your `INSTALLED_APPS`

```
INSTALLED_APPS = (  
    'inviter2'  
)
```

Include `inviter2.urls` into your root `urls.py` file under the `inviter2` namespace

```
urlpatterns = patterns(  
    # [...]  
    url('^invites/', include('inviter2.urls', namespace='inviter2')),  
)
```



---

## Usage

---

To invite people make use of `inviter2.utils.invite`

```
from inviter2.utils import invite

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

`inviter2.utils.invite` also allows you to make use of a custom email sending function, say to send HTML emails

```
from inviter2.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 inviter2 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 `inviter2.utils.invite` and `inviter2.utils.send_invite` for more information.

By default `inviter2.utils.send_invite` will render `inviter2/email/subject.txt` and `inviter2/email/body.txt` for the email.

`templates/inviter2/register.html` and `templates/inviter2/done.html` are rendered when registering respectively when done.

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



---

## Settings

---

There are a couple of editable settings

### **INVITER\_FORM**

**Default** `inviter2.forms.RegistrationForm`

**Type** `str`

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

### **INVITER\_FORM\_USER\_KWARG**

**Default** `instance`

**Type** `str`

The name of the kwarg field to pass to the `INVITER_FORM` with the user object.

### **INVITER\_REDIRECT**

**Default** `'inviter2: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** `'inviter2.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

### **INVITER\_FORM\_TEMPLATE**

**Default** `None`

Allows the user to specify a custom template for the inviter form.

### **INVITER\_DONE\_TEMPLATE**

**Default** `None`

Allows the user to specify a custom template for the done view.

**INVITER\_OPTOUT\_TEMPLATE**

**Default** None

Allows the user to specify a custom template for the opt-out view.

**INVITER\_OPTOUT\_DONE\_TEMPLATE**

**Default** None

Allows the user to specify a custom template for the opt-out done view.







I

INVITER\_DONE\_TEMPLATE, 9  
INVITER\_FORM, 9  
INVITER\_FORM\_TEMPLATE, 9  
INVITER\_FORM\_USER\_KWARG, 9  
INVITER\_FROM\_EMAIL, 9  
INVITER\_OPTOUT\_DONE\_TEMPLATE, 10  
INVITER\_OPTOUT\_TEMPLATE, 9  
INVITER\_REDIRECT, 9  
INVITER\_TOKEN\_GENERATOR, 9