

---

# kaleo Documentation

*Release 1.5*

**Eldarion**

October 06, 2014



<b>1</b>	<b>Development</b>	<b>3</b>
1.1	Contents .....	3



Provides a site with user to user invitations, working closely with *django-user-accounts* and requires Django 1.4.



The source repository can be found at <https://github.com/eldarion/kaleo>

## 1.1 Contents

### 1.1.1 Installation

- To install kaleo:

```
pip install kaleo
```

- Add `kaleo` to your `INSTALLED_APPS` setting:

```
INSTALLED_APPS = (  
    # other apps  
    "kaleo",  
)
```

- See the list of *Settings* to modify kaleo's default behavior and make adjustments for your website.
- Lastly you will want to add `kaleo.urls` to your urls definition:

```
...  
url(r"^invites/", include("kaleo.urls")),  
...
```

### Dependencies

#### **eldarion-ajax**

This is used to enable the markup based ajax. You can certainly wire up your own AJAX handling if you prefer to use something else. However, including this script will just have things work out of the box.

#### **pinax-theme-bootstrap**

This is semi-optional as the only reason it is required is the included `_invite_form.html` renders the form through the `as_bootstrap` filter. If you override this template in your project, you obviously remove this requirement at least in context of this app.

### django-user-accounts

Used for linking people to invitations when they confirm email addresses and sending and processing of signup codes via email.

#### 1.1.2 Usage

Integrating *kaleo* into your project is just a matter of using a couple of template tags and wiring up a bit of javascript. The invite form is intended to function via AJAX and as such returns JSON.

Firstly, you will want to add the following blocks in your templates where you want to expose the invite form and display to the user a list of who they have invited (of course you may choose to not expose that):

```
{% load kaleo_tags %}

<div class="invites">
  {% invite_form request.user %}

  <div class="sent">
    <h3>Invitations Sent</h3>
    {% invites_sent request.user %}
  </div>
</div>
```

Then if you had an account bar somewhere at the top of your screen where you showed the user if they were logged in or note you could have:

```
{% load kaleo_tags %}

{% invites_remaining user %}
```

You'll then need to include bootstrap-ajax:

```
<script src="{% static "js/bootstrap-ajax.js" %}"></script>
```

#### 1.1.3 Settings

##### KALEO\_DEFAULT\_EXPIRATION

**Default** 168 (7 days)

Defines the what will be the default expiration of invitations and is currently the only way to specify the expiration date of invitations.

##### KALEO\_DEFAULT\_INVITE\_ALLOCATION

**Default** 0

Defines the default allocation of invites that new users get when they signup. In order to enable unlimited invitations, just set this to -1.

#### 1.1.4 Templates

*kaleo* comes with two templates that are minimal snippets that get rendered from the template tags.



### **`_invite_form.html`**

This is a snippet that renders the form as well as a div to hold the contents of the response from the form AJAX submission.

### **`_invited.html`**

A simple unordered list of people you have invited that is linked to their profile when they join the site.

### **`_invites_remaining.html`**

A fragment to display how many invites a particular user has.

## **1.1.5 Signals**

Both of these signals are sent from *JoinInvitation* and provides a single keyword argument, *invitation* which is the relevant instance of *JoinInvitation*.

### **`kaleo.signals.invite_sent`**

This signal is sent immediately after the invitation is sent.

### **`kaleo.signals.invite_accepted`**

This signal is sent immediately after the acceptance of the invitation has been processed.

### **`kaleo.signals.joined_independently`**

This signal is sent when someone signs up using the same email address that exist for an invitation and they confirm that email address.

## **1.1.6 Management Commands**

### **`add_invites`**

Adds invites to all users with 0 invites remaining.

```
manage.py add_invites 10 # Adds 10 new invites to all users with 0 invites remaining
```

### **`infinite_invites`**

Gives all users unlimited invites

```
manage.py infinite_invites
```

## topoff\_invites

Makes sure all users have at least a certain number of invites.

```
manage.py topoff_invites 10 # Makes sure that all users have at least 10 invites
```

## 1.1.7 Views

There are four different views designed to handle POSTs via AJAX with a single variable, `amount`. These are designed to be able to help administrators manage invites from a front-end dashboard. The responses sent from these views conform to what `bootstrap-ajax` expects and works with.

They all require the user has the permission `kaleo.manage_invites` which is up to the site developer to determine how to and who to grant this to or evaluate in a custom auth backend. The largest use case should already be covered in that any user with staff or superuser privileges should supercede the need for this explicit permission.

### topoff\_all

```
url kaleo_topoff_all
```

Tops off all users with at least `amount` invites.

Returns:

```
{
  "inner-fragments": {
    ".invite-total": amount
  }
}
```

### topoff\_user

```
url kaleo_topoff_user user.pk
```

Tops off `{{ user.pk }}` with at least `amount` invites.

Returns:

```
{
  "html": amount
}
```

### addto\_all

```
url kaleo_addto_all
```

Adds `number` invites to all users

Returns:

```
{
  "inner-fragments": {
    ".amount-added": amount
  }
}
```

## addto\_user

**url** kaleo\_addto\_user user.pk

Adds number invites to `{{ user.pk }}`

Returns:

```
{
  "inner-fragments": {
    ".html": amount
  }
}
```

## invite\_stat

**url** kaleo\_invite\_stat user.pk

Returns a rendered `kaleo/_invite_stat.html` fragment to be supplied by the site developer to render an `InvitationStat` object for the `user.pk` provided to the template with the context variable `stat`.

The intended purpose for this is to be used as a `data-refresh-url` for `bootstrap-ajax`.

Returns:

```
{
  "html": <rendered kaleo/_invite_stat.html> # provided by site developer
}
```

## 1.1.8 CHANGELOG

### 1.2

- Django 1.5 custom user model support
- BACKWARD INCOMPATIBILITY: removed the import of `kaleo.receivers` from `urls.py`
- added a `joined_independently` signal
- refactored logic that existed in signal receivers to model methods
- fixed bug in add invites functionality that would set everyone to have unlimited invites
- fixed bug in top off invites functionality that removed unlimited invites from those who had it

### Backward Incompatibilities

Importing things into `models.py` or `urls.py` that have nothing to do with either of those modules has been a bit of a hack for awhile. Pinax is moving to a more explicit approach where these things are hooked up at the project level. In fact, Pinax starter projects will support `receivers.py` getting hooked up at runtime automatically through the use of a `startup.py` module that has code that gets executed in the `wsgi.py` and `manage.py`.

See how it all works in this commit:

<https://github.com/pinax/pinax-project-account/commit/364795cdd683574ab6a5093be34b9d47a3487bea>

### 1.1

- added a set of views to handle managing invites via ajax

### 1.0

- translated templates
- replaced *remaining\_invites* template tag with *invites\_remaining* inclusion template tag
- invite view is now bootstrap-ajax compatible

### 0.4

- moved away from Pinax dependencies and to require django-user-accounts
- moved to Django 1.4

### 0.3

- fixed documentation bugs
- added stats.py for stat collection

### 0.2

- added ability to set default invite allocation for new users
- added ability to enable users to have unlimited invitations

### 0.1

- initial release