

---

# **Connect Documentation**

*Release*

## **Organizing for Action**

April 01, 2016



<b>1 Administrator Guide</b>	<b>3</b>
<b>2 User Guide</b>	<b>29</b>
<b>3 Developer and Operator Guide</b>	<b>57</b>
<b>4 Contributing to Connect</b>	<b>83</b>
<b>5 Connect License</b>	<b>85</b>



Connect is a Python/Django-based communication platform built in-house by [Organizing for Action](#)

**Regular Users** looking for documentation on how to use Connect should refer to our [User Guide](#) section.

**Administrators** looking for documentation on how to use Connect should refer to our [Administrator Guide](#) section.

**Developers** looking for more information about installation and contributing to Connect can refer to the [Developer and Operator Guide](#) section.

Connect is available under the [MIT license](#) and the source code can be found at the official [Github repository](#) for third parties to use and contribute to. For more information about contributing view the [Contributing to Connect](#)



---

## Administrator Guide

---

This guide will cover features that are exclusive to all Connect administrators, including:

- Creating a new group
- Adding resources to groups
- Connect administrator features, accessed from the Admin menu, including:
  - Message moderation
  - Flag moderation log
  - Group moderation
  - Admin gallery
  - Invites
  - User reports
  - Group reports

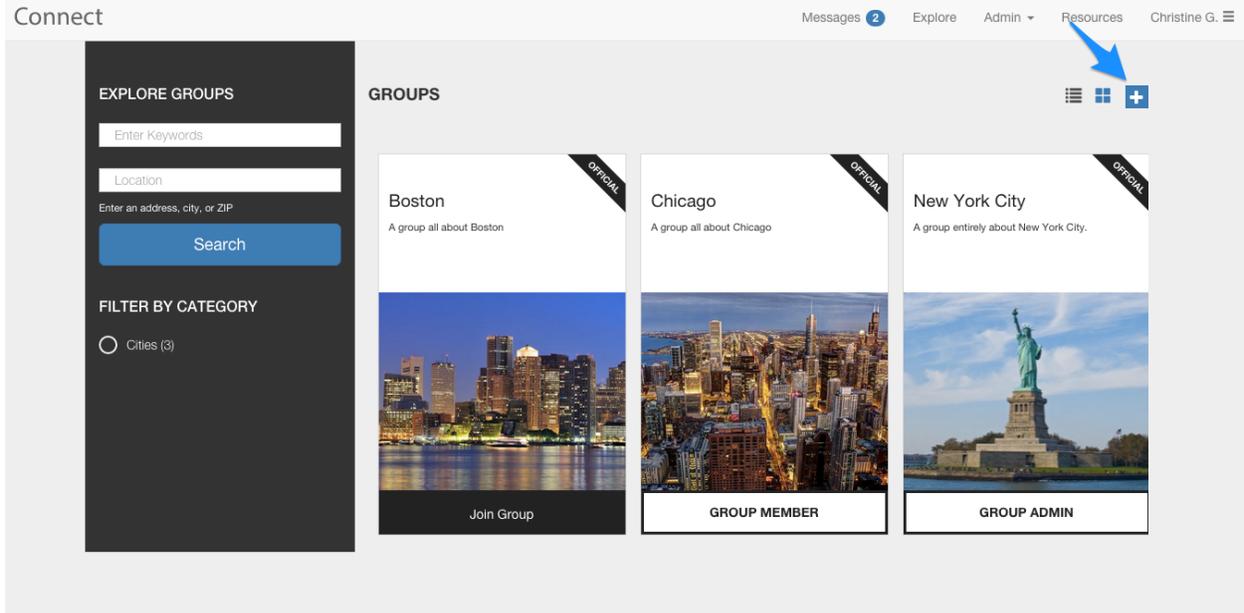
The guide will also cover features that are exclusive to Connect super-administrators, including managing other Connect users, and additional features accessed from the Admin menu:

- Popular URLs
- Category admin
- Tag admin

### 1.1 Creating A New Group

Connect administrators can create new groups by clicking the **Explore** option on the Connect navigation bar.

From the Groups page, click the plus (+) sign to create a new group.



On the **Create a Group** page, you can create and customize a new Connect group. While creating your group, give it a (1) name, category and description, and (2) a profile picture.

You'll also have the opportunity to add (3) additional group administrators, called **Owners** here, and (4) users who are whitelisted, whose messages will bypass moderation.

## Create a group

NAME\*

CATEGORY\*  1

DESCRIPTION

TAGS

STATE

GROUP IMAGE  Choose an image to use as the group's profile picture.  2

OWNERS  3 Type some text to sear

WHITELIST USERS  4 Type some text to sear These users will always be allowed to send to this group.

THIS IS AN OFFICIAL GROUP  
Official groups are managed by staff and appear first in search results.

PRIVATE  
Membership to private groups is moderated.

PUBLISH THIS GROUP  
Published groups can be seen by all users. Unpublished groups can only be seen if you have the link.

On this page, you'll also decide (1) whether or not the group is official (managed staff members of your organization), (2) if other users can join the group without being invited, (3) whether or not the group will appear to other Connect users, (4) if posts to the group are moderated, and (5) if other users can see who is a member of the group.



THIS GROUP HAS A LOCATION

MODERATE THIS GROUP  
Posts by users must be moderated by an admin.

MEMBER LIST PUBLISHED  
Group member list is public

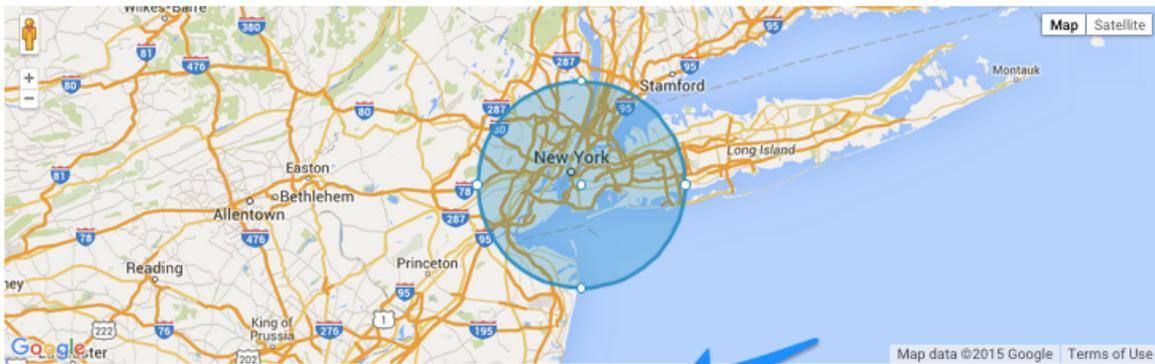
DISPLAY LOCATION

**2**

Optionally enter the location you'd like displayed on the group description page.

ADDRESS **1**

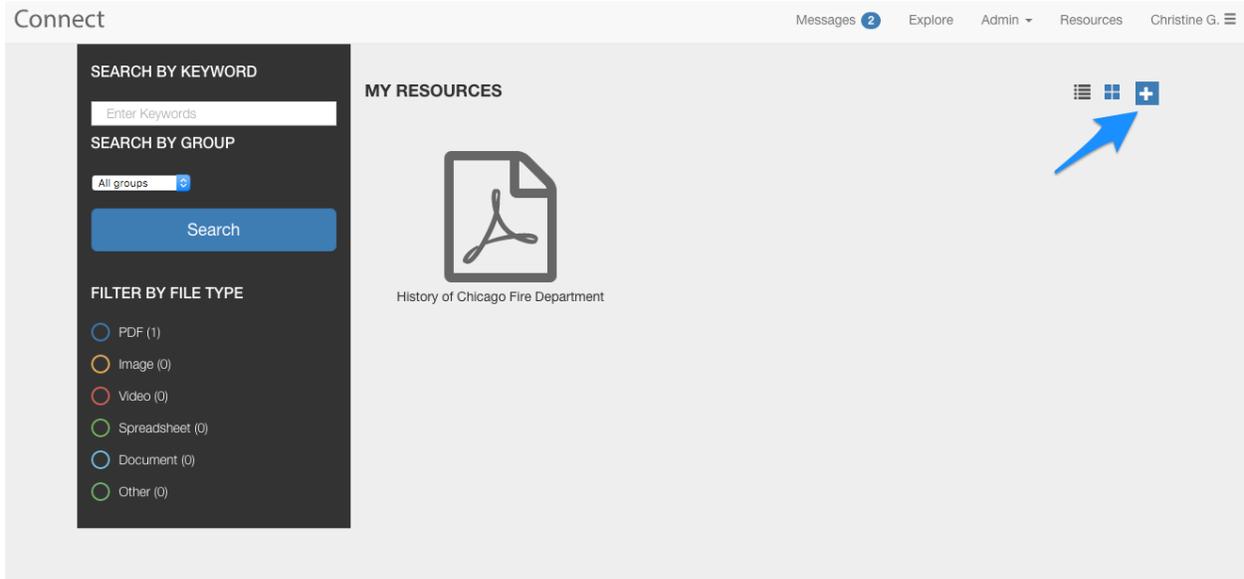
COVERAGE RADIUS, IN MILES



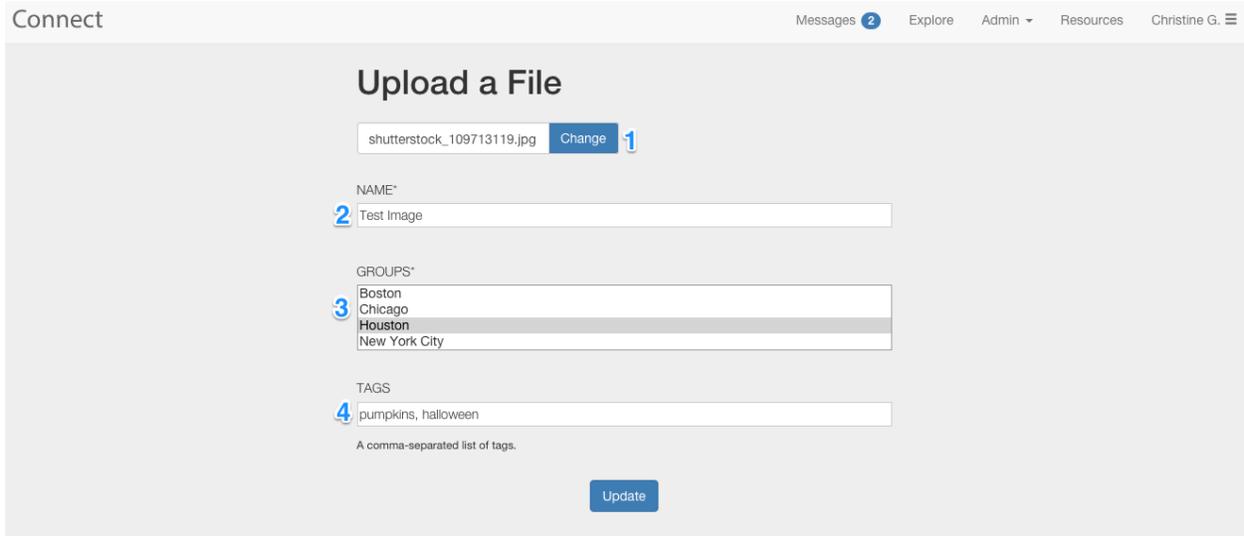
## 1.2 Resources

As a group administrator, you can add resources, such as pdf files, word documents, and images, to share with members of your groups by clicking the **Resources** option on the Connect navigation bar.

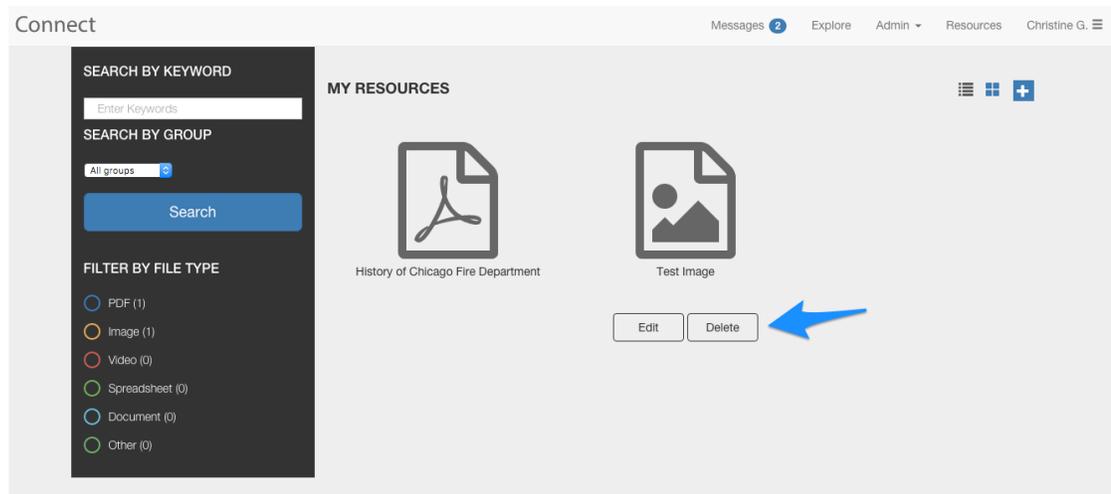
From the **Resources** page, click the plus (+) sign to add resources.



From the **Upload a File** page, (1) you can choose your file to upload, (2) give the file a name, (3) choose the group you'd like to share the file with, and (4) assign the file tags, or keywords, which will allow users to search for the file.



When you return to the **Resources** page, you'll be able to edit or delete any resources you've added.



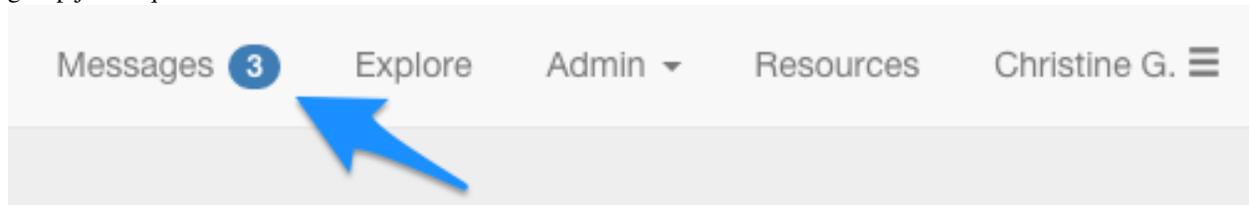
### 1.3 Message Moderation

In this section, we'll cover (1) finding messages to moderate and (2) moderating messages.

#### 1. Finding messages to moderate

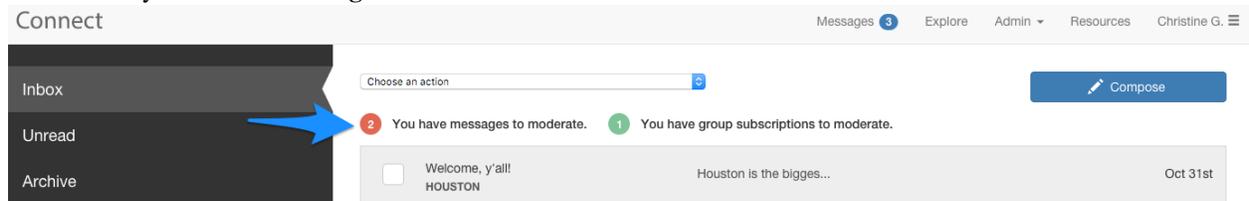
From any screen on Connect, the **Messages** option in the navigation bar will indicate the total number of messages available for you.

This number includes: unread messages from groups you're a member of, messages pending moderation, or pending group join requests.

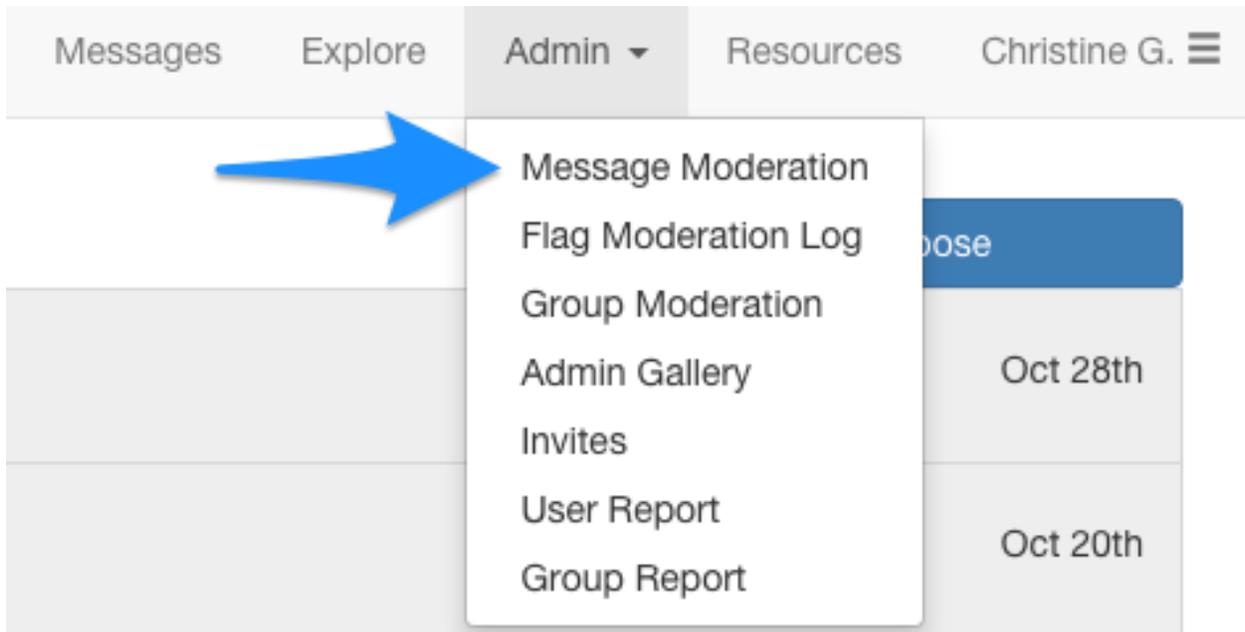


To access the **Message Moderation** panel, Connect administrators have two options:

1. Click the **Messages** from the Connect navigation bar to go to your Connect inbox. From there, click the link that says **You have messages to moderate**.



2. Choose **Message Moderation** from the Connect navigation bar



## 2. Moderating messages

From the **Message Moderation** panel, you'll see a preview of the messages available to moderate.

To see the full message pending, in context of the thread it was posted to, click the subject line of the message.

Connect Messages 4 Explore Admin Resources Christine G. ☰

### Message Moderation

Moderate All (3 Messages To Moderate)  
Moderate Houston (3 Messages To Moderate)

Status	Sender	Thread Subject & Group	Snippet	Actions (Approve   Veto   Reset)
Pending	Audrey R	Welcome, y'all! Houston	Wow, I had no idea Ho...	<input type="radio"/> Approve <input type="radio"/> Veto
Pending	Audrey R	Best neighborhoods in Houston? Houston	Hi everyone, I'm movin...	<input type="radio"/> Approve <input type="radio"/> Veto
Pending	Audrey R	Hello! Houston	So glad to see this C...	<input type="radio"/> Approve <input type="radio"/> Veto

[Save changes](#)

Now you can read the entire message, and where it fits within the thread. Click the arrows at the bottom of the page to expand and contract the view of the entire thread.

Connect Messages **4** Explore Admin ▾ Resources Christine G. ☰

[Back to Moderation](#)

**HOUSTON**

**Message from Thread: Welcome, y'all!**

This message is pending approval.

**Audrey R.**  
Today at 3:34 PM

Wow, I had no idea Houston is that big! How soon do you think it will be before it overtakes Chicago in population size?

Here is the message in the thread context:

**Christine G.** Houston is the bigges... Today at 2:34 PM

---

**Audrey R.** Wow, I had no idea Ho... Today at 3:34 PM



Back in the **Message Moderation** panel, you can choose to **Approve** or **Veto** (reject) messages from your group in two ways. (1) Click the **Approve** or **Veto** button associated with each message, or, (2) moderate all messages in bulk by clicking the **Approve** or **Veto** options next to Actions. Then click (3) **Save changes**.

Connect Messages **4** Explore Admin ▾ Resources Christine G. ☰

## Message Moderation

Moderate All (3 Messages To Moderate)

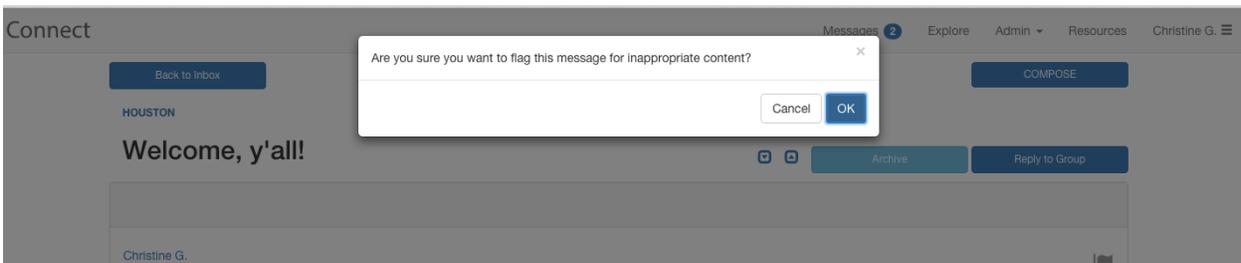
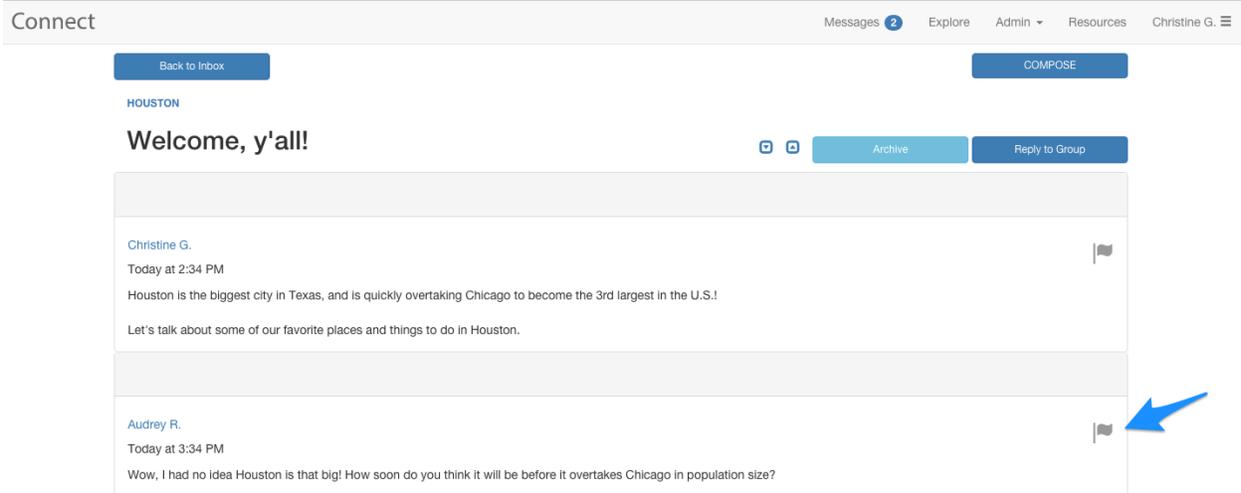
Moderate Houston (3 Messages To Moderate)

Status	Sender	Thread Subject & Group	Snippet	2 Actions (Approve   Veto   Reset)
Pending	Audrey R	<b>Welcome, y'all!</b> Houston	Wow, I had no idea Ho...	1 <input type="radio"/> Approve <input type="radio"/> Veto
Pending	Audrey R	<b>Best neighborhoods in Houston?</b> Houston	Hi everyone,I'm movin...	<input type="radio"/> Approve <input type="radio"/> Veto
Pending	Audrey R	<b>Hello!</b> Houston	So glad to see this C...	<input type="radio"/> Approve <input type="radio"/> Veto

[Save changes](#) **3**

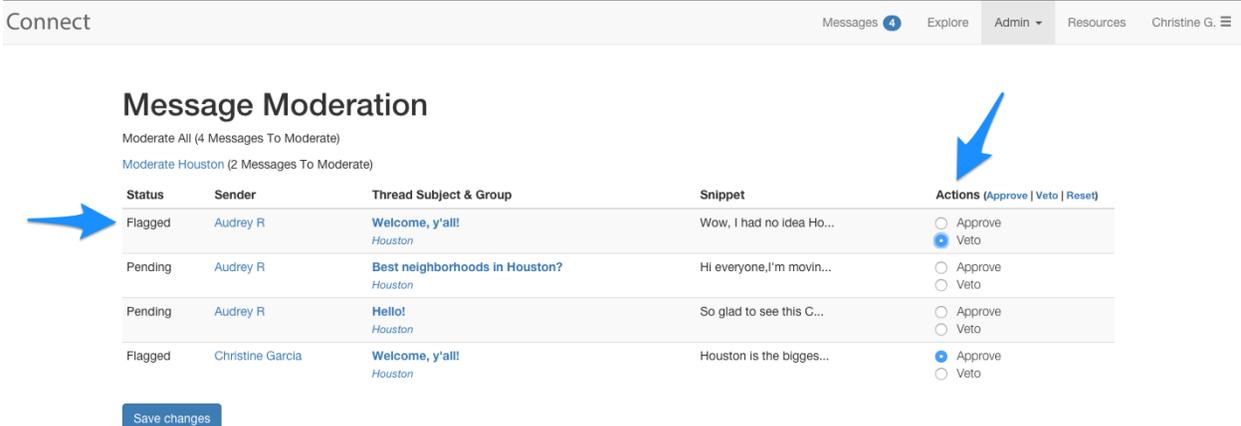
## 1.4 Flag Moderation Log

Any member of a group can flag messages as inappropriate by clicking the flag icon located on the bottom right corner of a message.

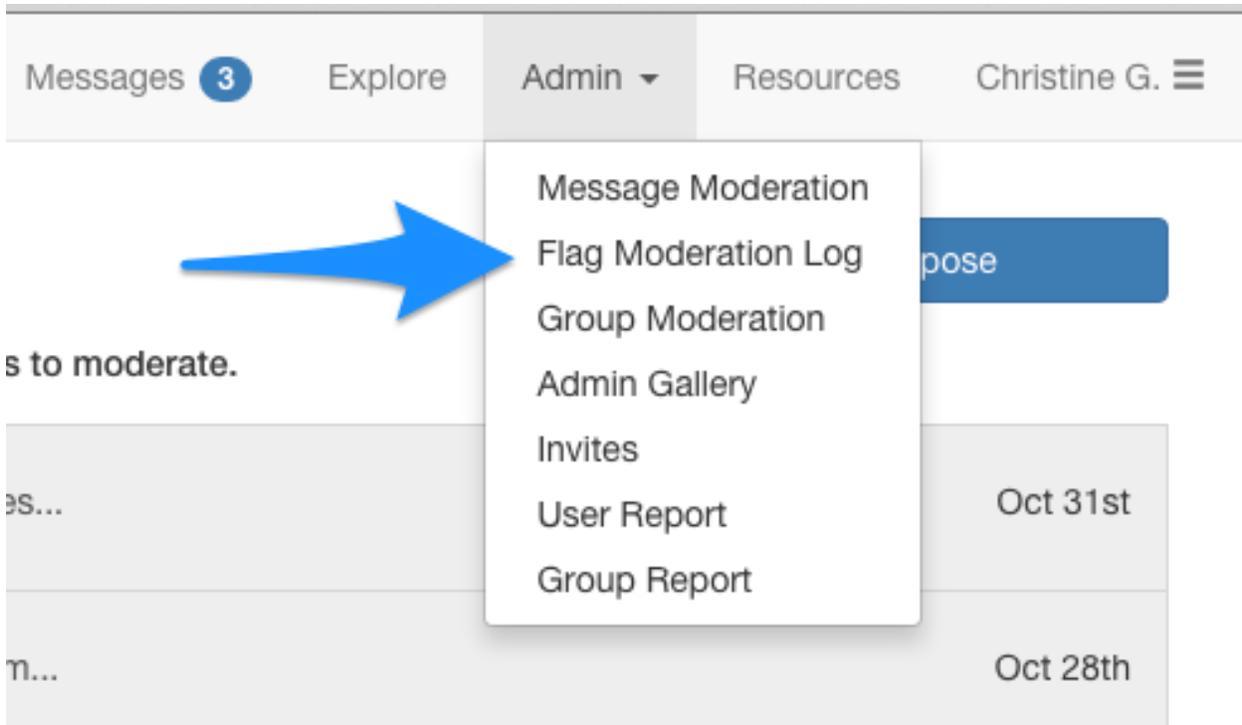


Once the message is flagged, it is removed from view by all group members, except the member who originally posted the message.

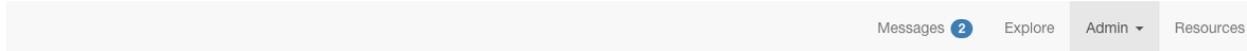
Flagged messages will appear in your **Message Moderation** panel for further review. If you decide to veto a message, it will be permanently removed from the group. If you choose to approve the message, it will be returned to the group.



When a flagged message is either approved or vetoed, it will appear in your **Flag Moderation Log**. You can access the **Flag Moderation Log** from the **Admin** menu on the Connect navigation bar.



In the **Flag Moderation Log**, you can see (1) the subject line of the flagged message, (2) who flagged the message and when, (3) who moderated the message and when, and (4) the action taken on the message.



## Flag Log

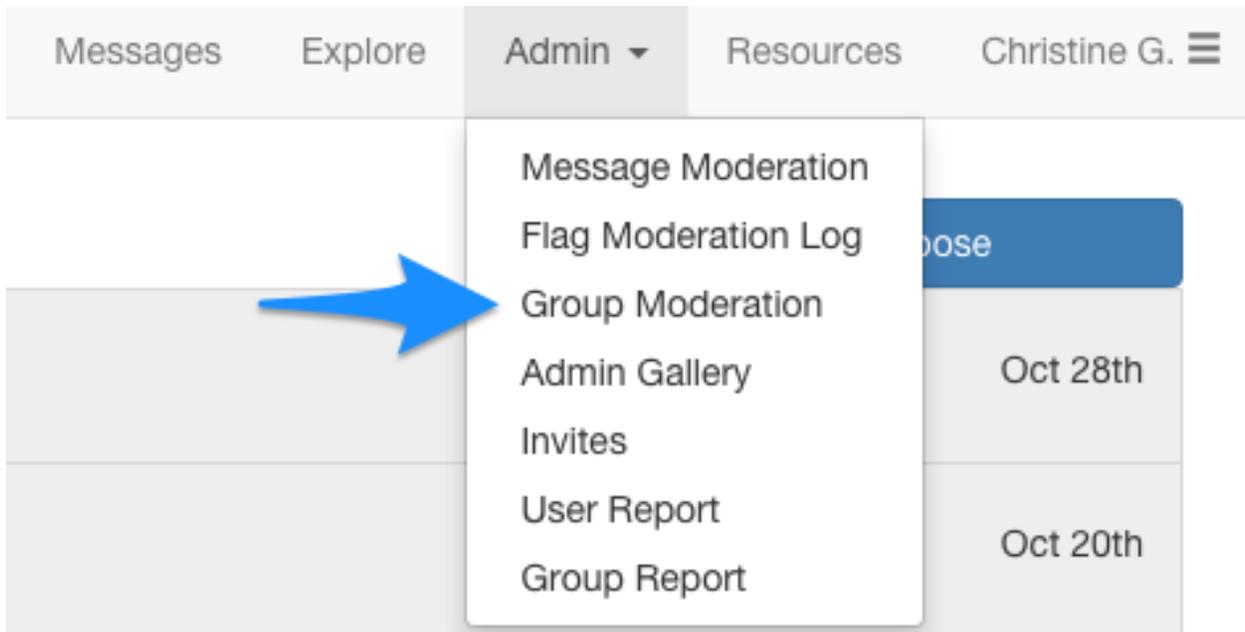
From:  Until:  Per Page: 25

Message <b>1</b>	Current status	Flagged by <b>2</b>	Flagged at	Moderated by <b>3</b>	Moderated at	Action <b>4</b>
Houston Is the biggest...	approved	Audrey R.	Oct. 31, 2015, 3:57 p.m.	Christine G.	Oct. 31, 2015, 3:59 p.m.	approved
Wow, I had no idea Ho...	vetoed	Christine G.	Oct. 31, 2015, 3:55 p.m.	Christine G.	Oct. 31, 2015, 3:59 p.m.	vetoed
What's the best time...	vetoed	Nick C.	Oct. 22, 2015, 10:47 p.m.	Nick C.	Oct. 22, 2015, 10:47 p.m.	vetoed
	approved	Nick C.	Oct. 20, 2015, 4:51 p.m.	Nick C.	Oct. 20, 2015, 4:52 p.m.	approved

The **Flag Moderation Log** is especially helpful if you have multiple administrators in a single group.

## 1.5 Group Moderation

If your group is private, users must request to join your group. You can access the **Group Moderation** panel from the **Admin** menu on the Connect navigation bar.



From the Moderate your join requests panel, you can select individual or all pending group join requests. Select the users whose join requests you'd like to approve or reject, and click **Save changes**.

## Moderate your join requests

Open requests — [Toggle all](#)

- Christine OpenConnect (christine.celeste+openconnect@gmail.com / , ) requested to join Houston.
- Christine Garcia (christine.celeste+openconnect2@gmail.com / , ) requested to join Houston.

Action

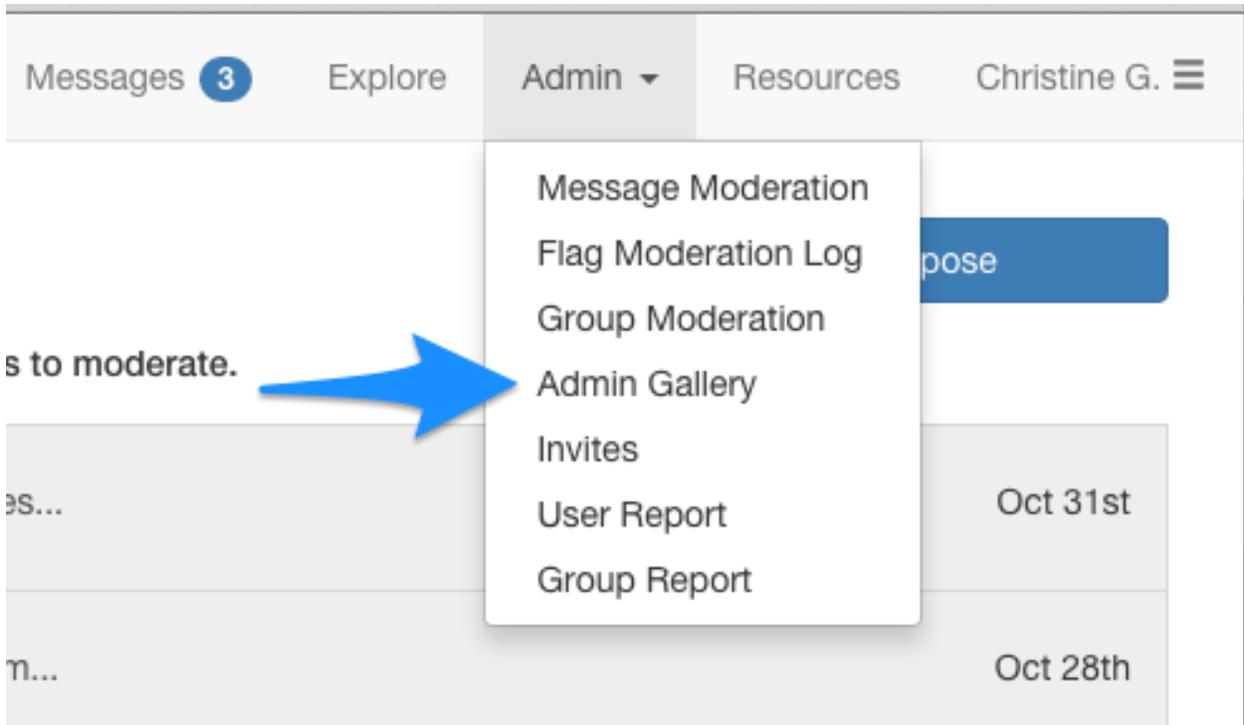
Approve

Reject

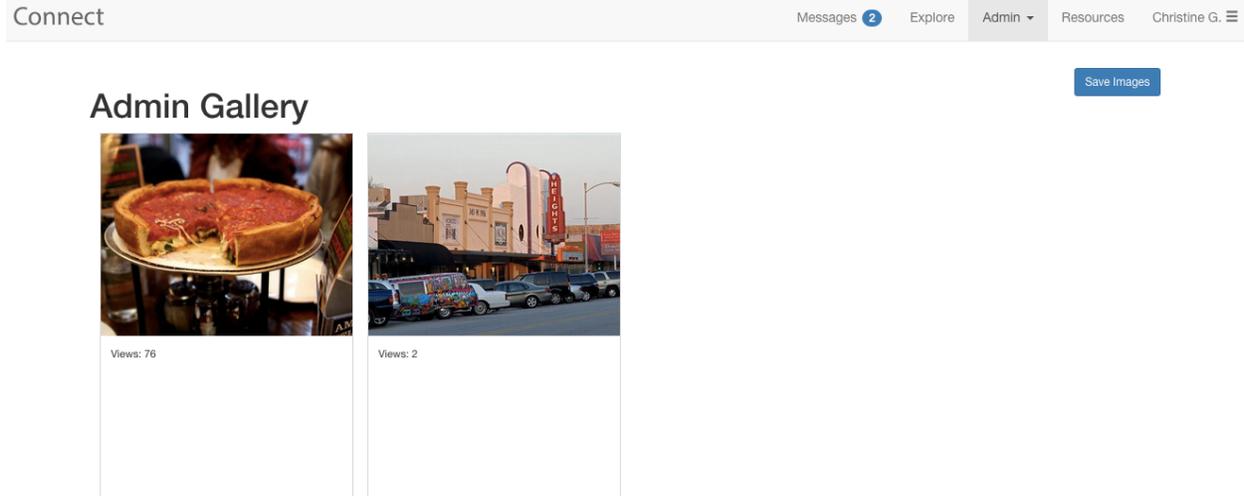
[Save changes](#)

### 1.6 Admin Gallery

Access the **Admin Gallery** panel from the **Admin** menu on the Connect navigation bar.



In the **Admin Gallery**, you can find images that have been shared in Connect groups, and see how many times each has been viewed.



You can also (1) select images individually, or select all images, and (2) save them to a Dropbox account.

Connect

Messages **2**

Explore

Admin ▾

Resources

Christine G. ☰

## Admin Gallery

1 Select images to save to your Dropbox account: [Select All](#) [Select None](#)

2 [Save to Dropbox](#)



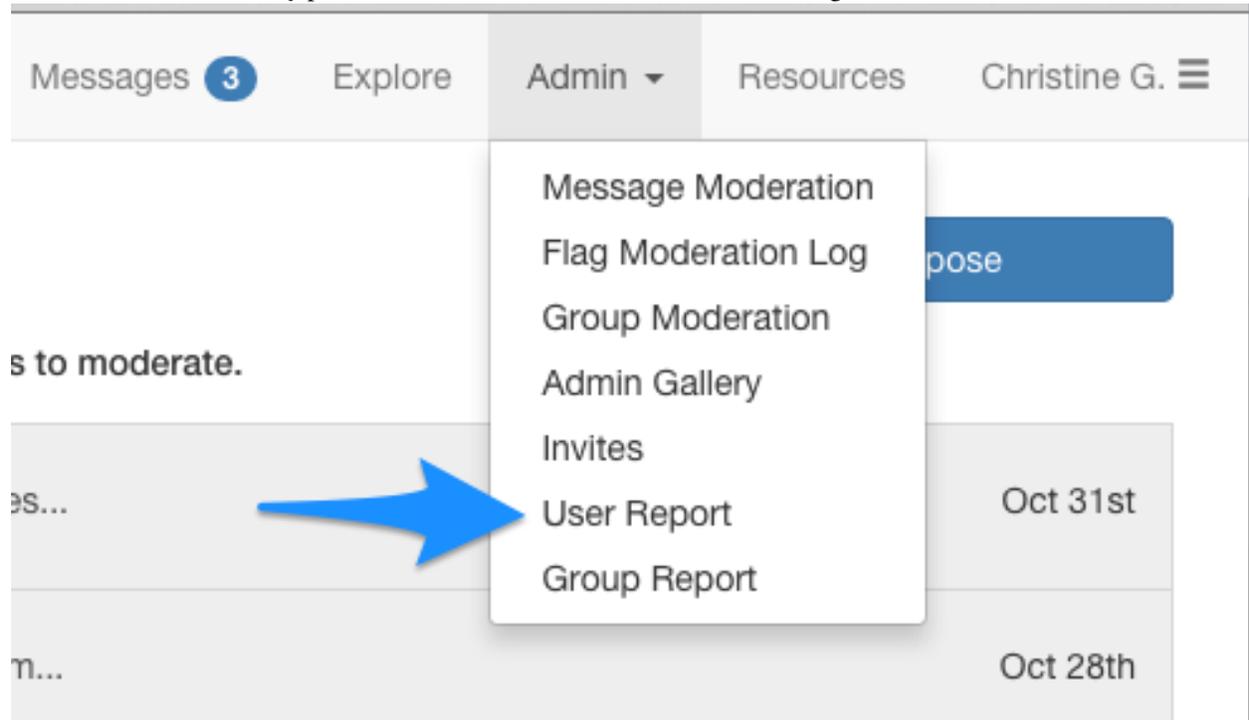
Save to Dropbox?  
Views: 76



Save to Dropbox?  
Views: 2

## 1.7 User Report

Access the **Admin Gallery** panel from the **Admin** menu on the Connect navigation bar.



In the **User Report**, you can find information on who has joined Connect. You can access the **User Report** panel from the **Admin** menu on the Connect navigation bar.

In the **User Report**, you can find information on when individual users including:

- The user's first name and last initial
- The user's email address

- When the user joined Connect
- The user's last login date
- The number groups the user has joined
- The number of messages the user has sent
- The number of messages the user has flagged
- The number of times the user has visited the site.
- The user's current status on Connect—If they have been banned or have unsubscribed since joining

You can also export this information as a .csv file.

Connect Messages 4 Explore Admin Resources Christine G. ☰

### User Report

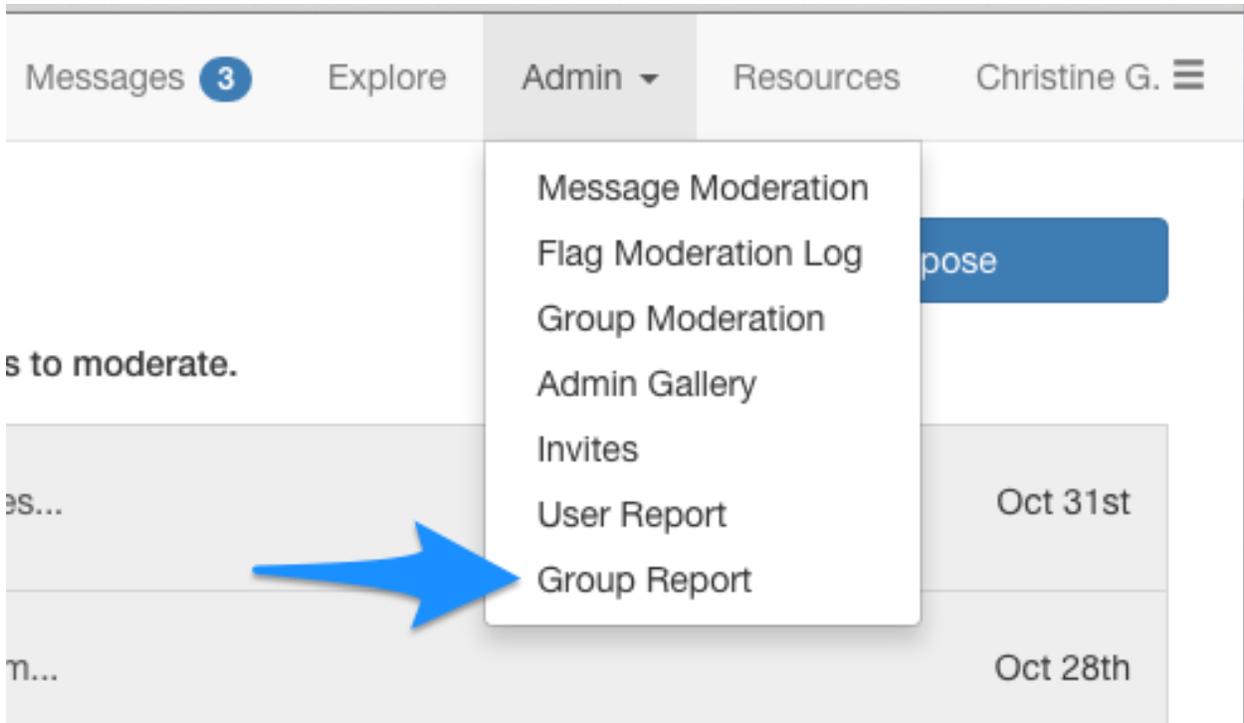
Name:  From:  Until:  Per Page: 25

[Search](#) [Export](#)

Name	Email	Phone	Zip	State	Joined	Last Login	Groups Joined	Flags received	Messages sent	Visits	Banned	Unsubscribed	Staff	Superuser
Nick C.	[REDACTED]				Oct. 18, 2015, 6:09 p.m.	Oct. 28, 2015, 4:12 p.m.	3	1	4	11			☑	☑
Brock D.	[REDACTED]				Oct. 19, 2015, 10:01 a.m.	Oct. 19, 2015, 10:01 a.m.	2	0	0	1				
Jonathan K.	[REDACTED]				Oct. 19, 2015, 11:07 a.m.	Oct. 19, 2015, 11:07 a.m.	1	0	0	1				
Christine G.	[REDACTED]				Oct. 19, 2015,	Oct. 19, 2015,	3	2	4	7				

## 1.8 Group Report

Similar to the **User Report**, in the **Group Report** you can find information about groups on Connect. You can access the **Group Report** panel from the **Admin** menu on the Connect navigation bar.



In the **Group Report**, you can find information on groups including:

- Names of groups
- The total number of messages sent within the group
- The number of threads started in the group
- The number of replies
- The number of users posting to the group
- The group category and location
- The total number of group members
- The privacy level of the group
- The original creator of the group

You can also export this information as a .csv file.

## Group Report

From:  Until:  Per Page: 25 ▾

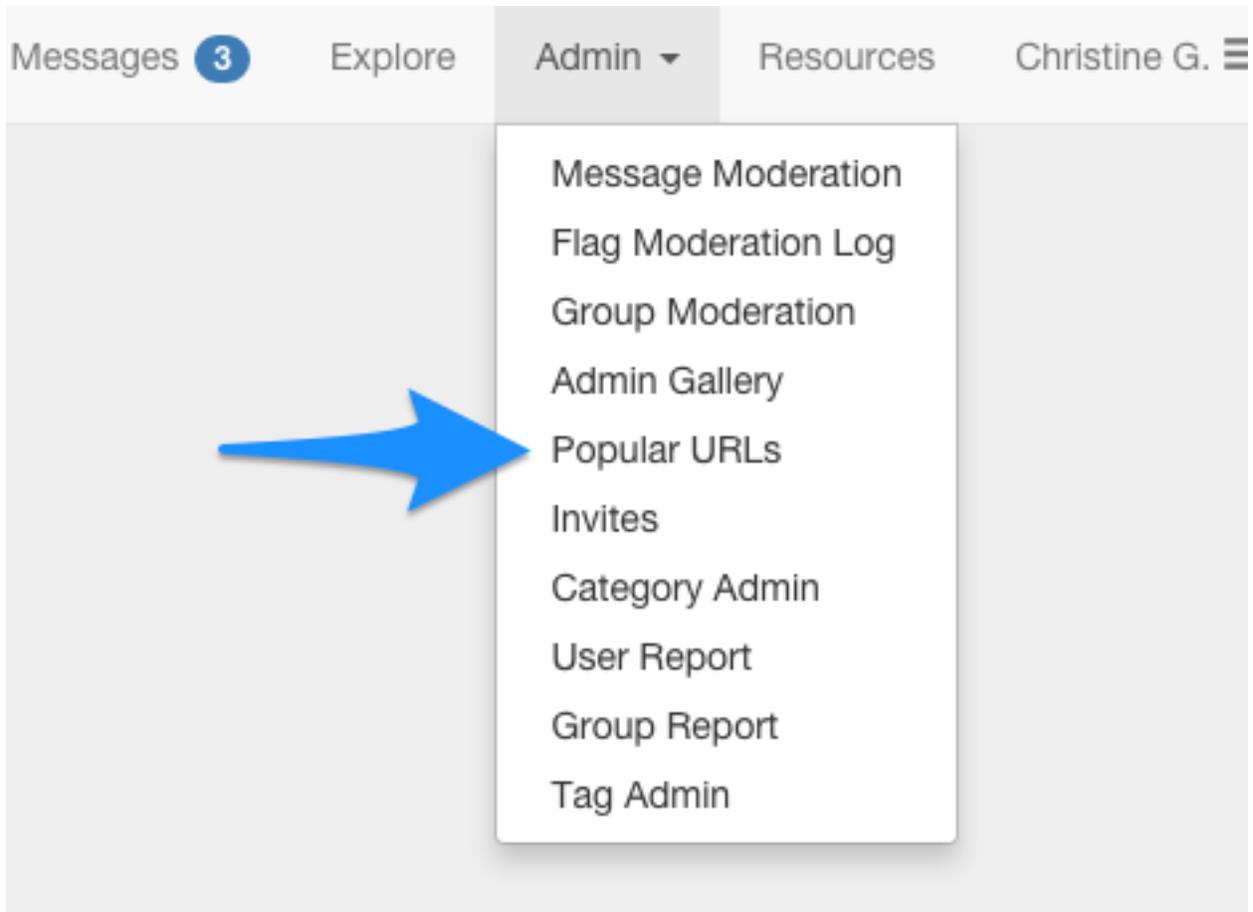
Update export

Name	Messages	Threads	Replies	Posters	Flagged messages	Category	Tags	State	Members	Admins	Private	Published	Moderated	Featured	Member List Published	Created
Chicago	5	2	3	3	2	Cities		IL	9	1		Yes		Yes	Yes	Oct. 18, 2015, 11:16 p.m.
New York City	2	1	1	2	0	Cities		NY	6	3		Yes		Yes	Yes	Oct. 22, 2015, 10:37 p.m.
Boston	0	0	0	0	0	Cities		MA	2	2		Yes		Yes	Yes	Oct. 28, 2015, 4:33 p.m.
Houston	5	3	2	2	2	Cities		TX	2	1	Yes	Yes	Yes	Yes	Yes	Oct. 31, 2015, 1:31 p.m.

## 1.9 Popular URLs

### Super Administrators Only

If you're interested in finding the top URLs being shared across Connect, you can access a **Popular URLs** report from the **Admin** menu on the Connect navigation bar.



From the **Popular URLs** report, you can sort URLs by create date, the number of messages associated with the URL, and the number of clicks the URL has received.

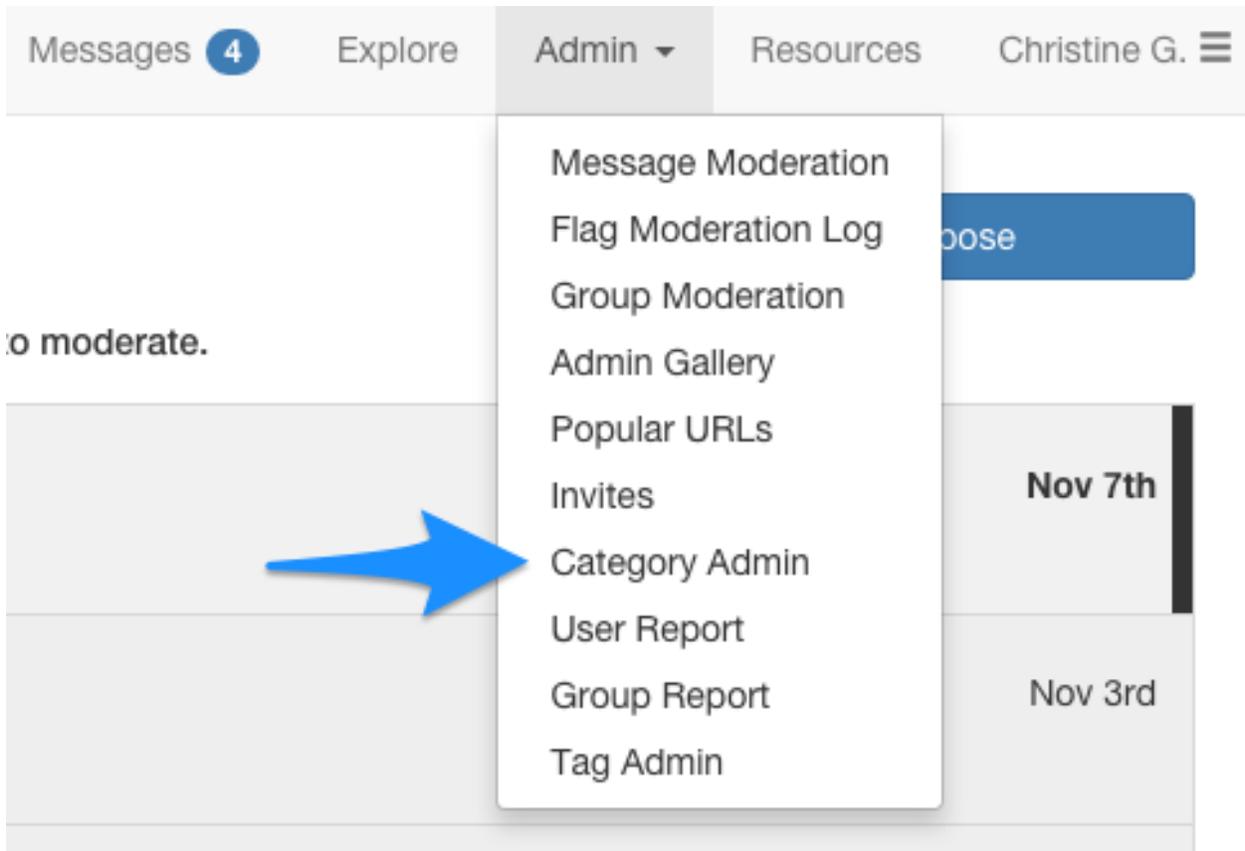
## Link popularity

Created <sup>1</sup>	Source	Short URL	Message Count <sup>2</sup>	Clicks <sup>3</sup>
Oct. 22, 2015, 10:46 p.m.	<a href="http://www.metmuseum.org/">http://www.metmuseum.org/</a>	MQ	3	1
Nov. 3, 2015, 9:03 p.m.	<a href="http://www.mfah.org/">http://www.mfah.org/</a>	Mg	1	1
Nov. 3, 2015, 9:03 p.m.	<a href="https://www.menil.org/">https://www.menil.org/</a>	Mw	1	3

## 1.10 Category Admin

### Super Administrators Only

In the **Category Admin** feature, you can add more categories for Connect groups. **Category Admin** is accessible from the **Admin** menu on the Connect navigation bar.



In **Category Admin**, you can add new categories, edit existing categories, and find information on the editing history of categories.

To (1) edit an existing category, simply click the name of the category to access the editing panel. To (2) add a new Connect category, click the **Add Category** button. And, if you'd like to (3) delete a category, click the box next to the category name, then click **Delete selected categories** from the **Action** dropdown.



When you click the name of a category to change it, you can also view the editing history of that category.



Here you find out when a change action was taken, by which user, and the type of action taken.

Django administration Welcome, Christine. View site / Log out

Home > Groups > Categories > Cities > History

**Change history: Cities**

Date/time	User	Action
Oct. 22, 2015, 10:32 p.m.	nick@connect.com (Nick C.)	
Nov. 3, 2015, 9:49 p.m.	christine@connect.com (Christine G.)	Changed name.
Nov. 3, 2015, 9:50 p.m.	christine@connect.com (Christine G.)	Changed name.

To add a new category, from the **Category Admin** panel, click **Add Category**.

You'll need to first (1) choose a slug for your group. The slug assigns a name to the exact location of your group within the site. Then, (2) choose a name for your category, to be displayed on the **Explore** page. You can also (3) adjust color of the category name by assigning a color using hexadecimals.

Then click **Save** to continue editing, add more categories, or exit to the **Category Admin** panel.

Django administration Welcome, Christine. View site / Log out

Home > Groups > Categories > Add Category

**Add Category**

Slug: 1 regions

Name: 2 Regions

Category Color: #000000 3

Modified at: (None)

Created at: (None)

After you add a new category, you'll need to add a group to your new category for it to appear in **Explore**.

Connect Messages 4 Explore Admin Resources Christine G. ☰

### Create a group

NAME\*  
West Coast

- Default
- Cities
- States
- Regions**

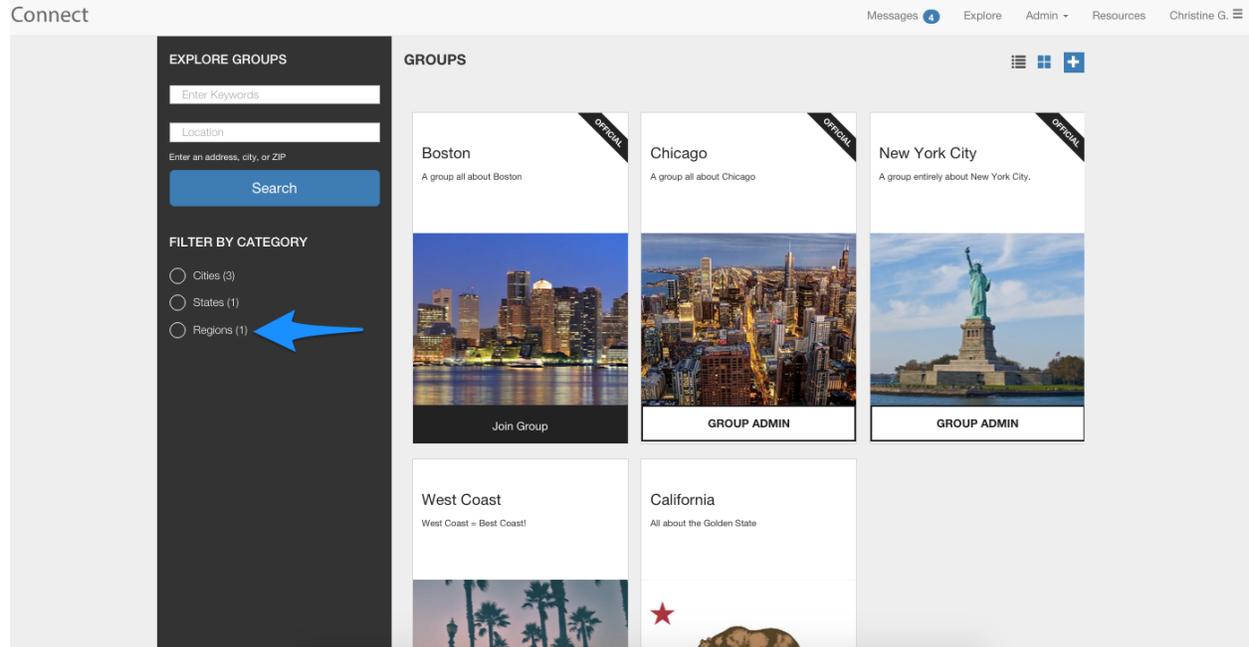
DESCRIPTION  
B I ∞  
West Coast = Best Coast!

GROUP IMAGE  
Choose an image to use as the group's profile picture.

OWNERS

WHITELIST USERS  
 These users will always be allowed to send to this group.

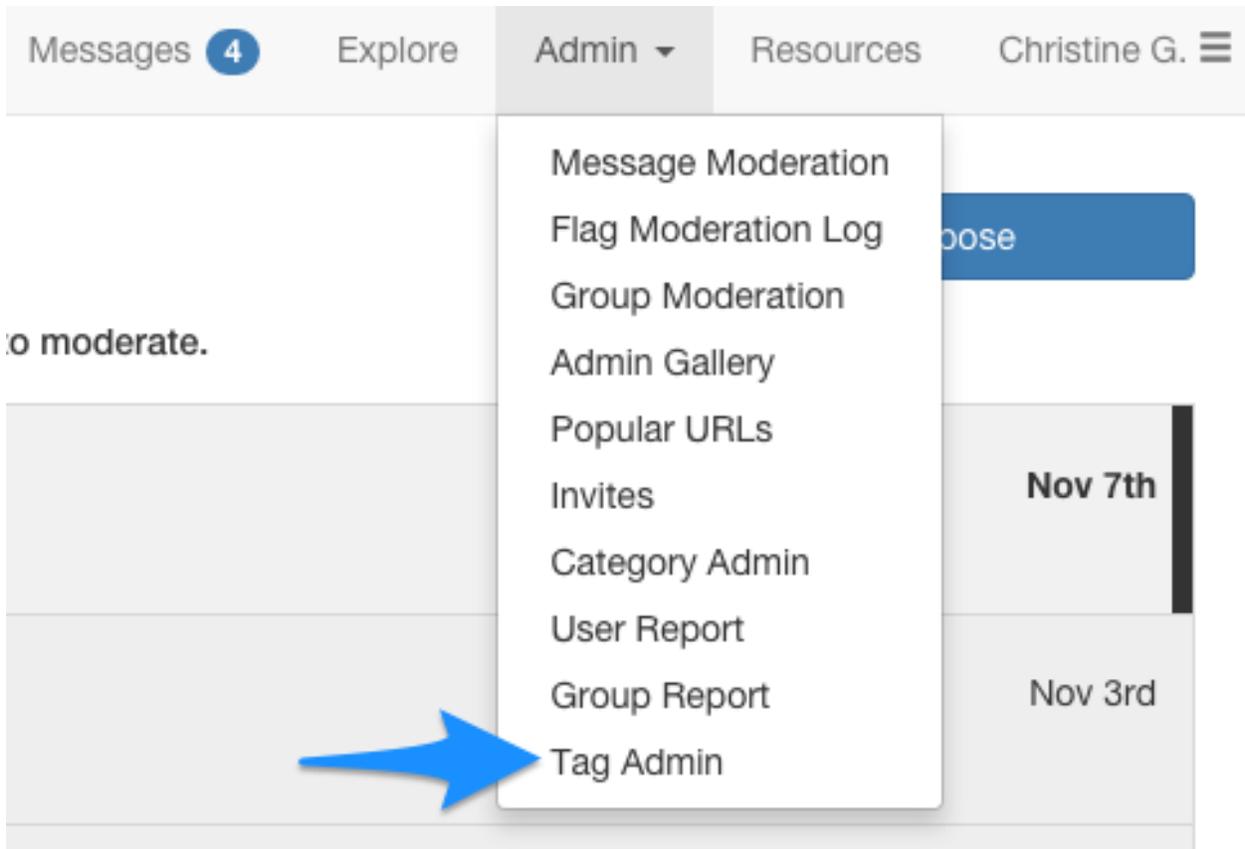
Now that you've created a group in the new category, you can find it in the **Explore** page.



## 1.11 Tag Admin

### Super Administrators Only

If you'd like to change the tags associated with the files in your resources section, or add new tags, you can do that in the **Tag Admin** feature. **Tag Admin** is accessible from the **Admin** menu on the Connect navigation bar.



To (1) edit an existing tag, click the name of the tag to access the editing panel. To (2) add a new tag, click the **Add Tag** button. And, if you'd like to (3) delete a tag, click the box next to the tag name, then click **Delete selected tags** from the **Action** dropdown.



By clicking an existing tag, you can edit the name of the tag and find the editing history of the tag.



In the **History** section, you find out when a change action was taken, by which user, and the type of action taken.



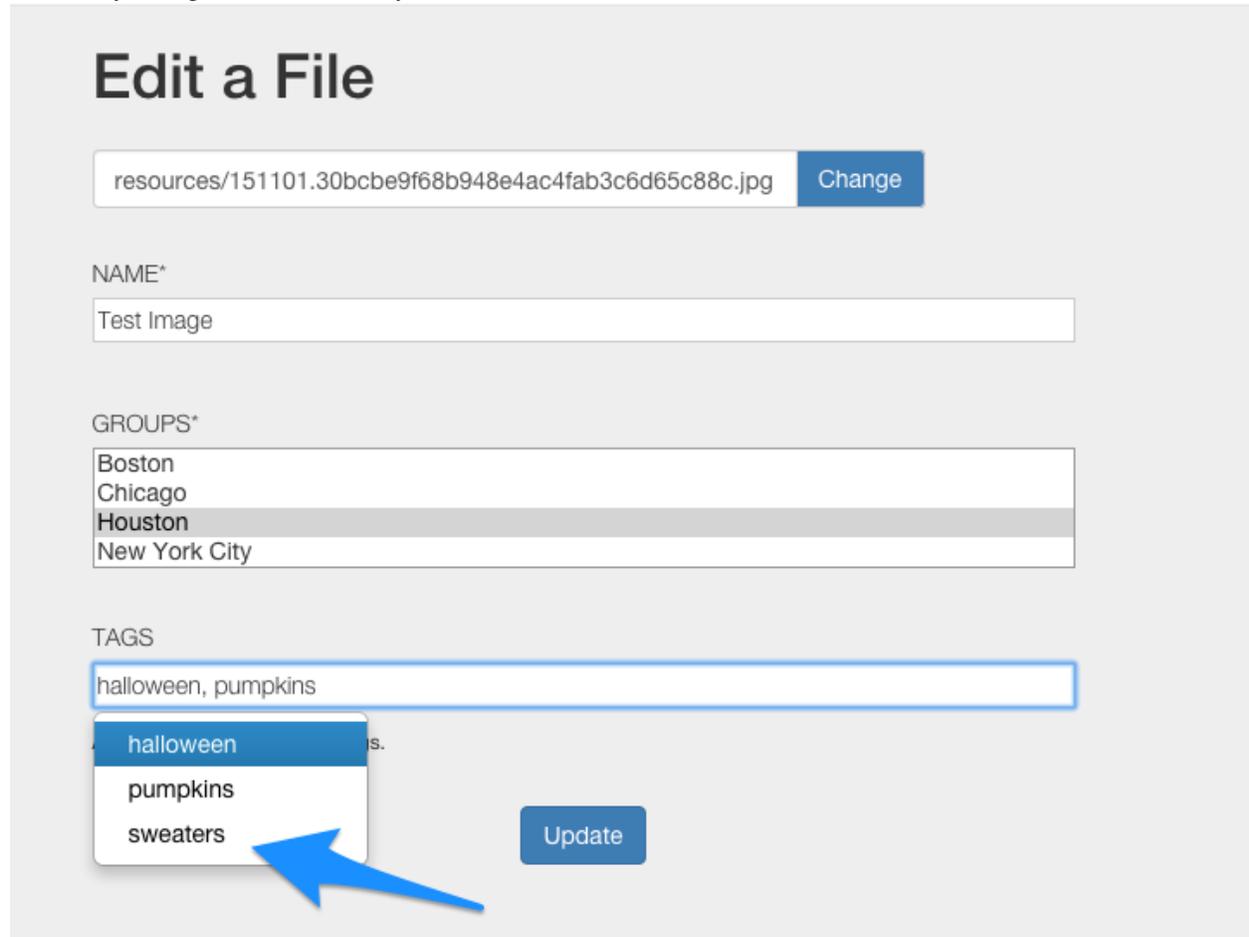
To add a new tag, from the **Tag Admin** panel, click **Add Tag**.

You'll need to first (1) choose a slug for your tag. The slug assigns a name to the exact location of your tag within the site. Then, (2) choose a name for your tag, to be displayed with the resources associated with your tag. Then click **Save** to continue editing, add more categories, or exit to the **Tag Admin** panel.



The screenshot shows the Django administration interface for adding a tag. At the top, it says 'Django administration' and 'Welcome, Christine. View site / Log out'. Below that is a breadcrumb trail: 'Home > Taggit > Tags > Add Tag'. The main heading is 'Add Tag'. There are two input fields: 'Name:' with a text box and a small icon to its right, and 'Slug:' with a text box. Below the form is a 'Cancel' link.

Now that your tag has been created, you can add it to Connect resources.



The screenshot shows the 'Edit a File' form. At the top, there's a text box containing the file path 'resources/151101.30bcbe9f68b948e4ac4fab3c6d65c88c.jpg' and a 'Change' button. Below that is a 'NAME\*' field with 'Test Image' entered. Underneath is a 'GROUPS\*' dropdown menu with options: 'Boston', 'Chicago', 'Houston' (which is selected and highlighted), and 'New York City'. Below the groups is a 'TAGS' field containing 'halloween, pumpkins'. A dropdown menu is open below the tags field, showing options: 'halloween', 'pumpkins', and 'sweaters'. A blue arrow points to the 'sweaters' option. To the right of the tags dropdown is an 'Update' button.

## 1.12 Managing Other Connect Users

### Super Administrators Only

If you have super-administrator privileges on Connect, you can manage the profiles of other Connect users. This includes (1) banning the user, (2) becoming the user, (3) updating the user's profile, and (4) updating the user's permissions.



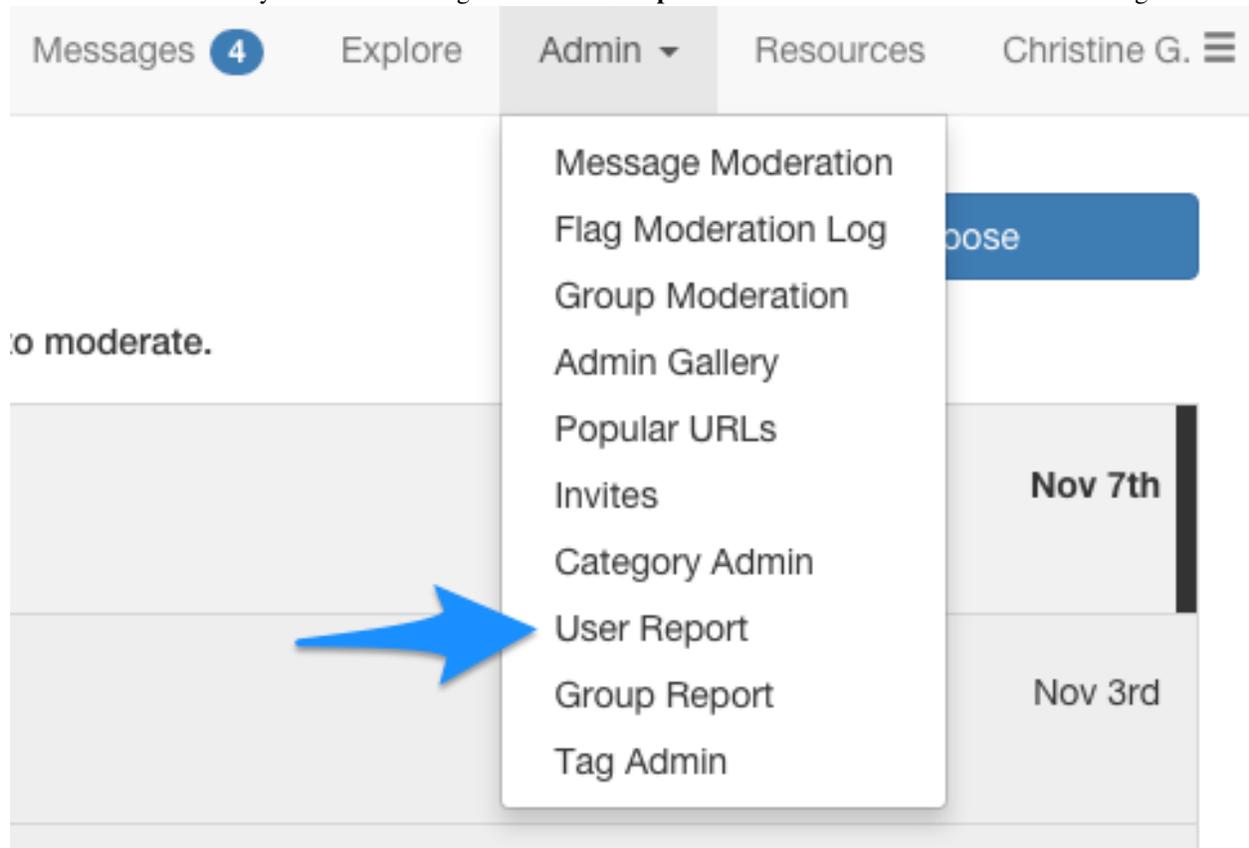
Alvin F.

Moved to NYC a few months ago, and lovin' every minute of it!

- Send a message
- 1 Ban User
- 2 Become User
- 3 Update user
- 4 Update Permissions

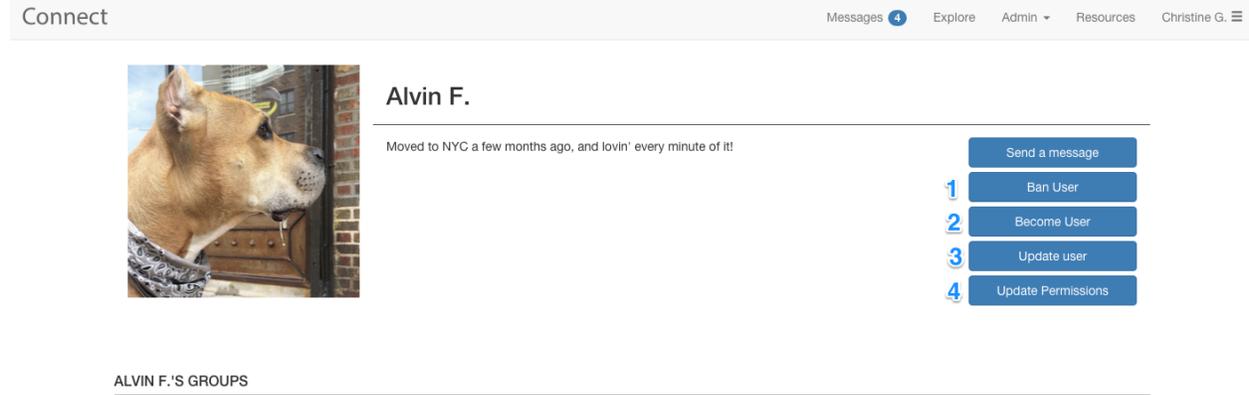
ALVIN F.'S GROUPS

Find the Connect users you'd like to manage via the **User Report** in the **Admin** menu of the Connect navigation bar.



From the **User Report**, search for the user whose account you'd like to manage, and click on their name. ... image:: images/image10.png

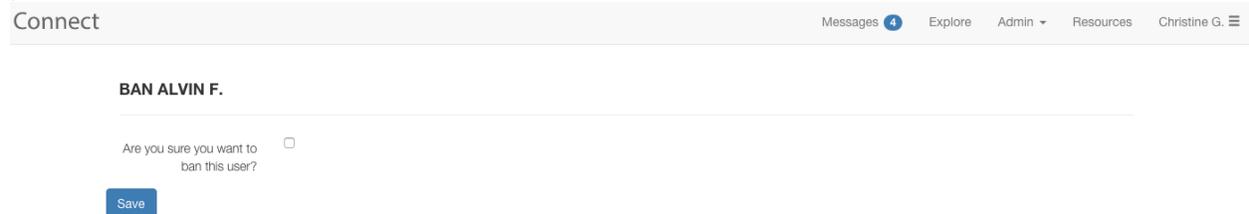
Once you have accessed a user's profile, you can take the steps mentioned above: **Ban User**, **Become User**, **Update User** and **Update Permissions**.



The screenshot shows the user profile for "Alvin F." in the Connect application. On the left is a profile picture of a dog. To the right of the picture, the name "Alvin F." is displayed, followed by a bio: "Moved to NYC a few months ago, and lovin' every minute of it!". Below the bio is a vertical list of four blue buttons: "Send a message", "Ban User", "Become User", "Update user", and "Update Permissions". The "Ban User" button is highlighted with a blue number "1" to its left. Below the buttons, the text "ALVIN F.'S GROUPS" is visible.

### 1.12.1 Ban User

Click (1) **Ban User** to ban the user from Connect. When you ban a user, their messages will only be visible to themselves –other Connect users will no longer see their messages. To ban a user, click the box next to **Are you sure you want to ban this user?** and click **Save**.



The screenshot shows a confirmation dialog titled "BAN ALVIN F.". It contains the text "Are you sure you want to ban this user?" followed by an unchecked checkbox. Below the text is a blue "Save" button.

### 1.12.2 Become User

If you'd like to find out more information about a user's activity on Connect, click (2) **Become User** to view Connect as the user would.



The screenshot shows a confirmation dialog titled "BECOME". It contains the text: "This will log you in as christine.celeste+openconnect@gmail.com. Everything on the site will appear as it does to them except for a banner reminding you that you are viewing the site as someone else." Below the text is a blue "Agree" button.

You will see a red banner informing you of which user's account you're current viewing Connect as, and a link to return to your own account.

You can now view the user's Connect messages and profile, as the user would. From the user profile, you can also edit their profile information.

Connect Messages 2 Explore Resources Alvin F. ☰

You are currently viewing the site as [redacted]@gmail.com. [Click here to return to your account.](#)

### Alvin F.

Moved to NYC a few months ago, and lovin' every minute of it!

[Edit My Profile](#)

#### MY SUBSCRIPTIONS

Houston

**Houston**

Don't send email notifications

Send a daily digest

Send me an email for every new message

[Save changes](#)

#### MY GROUPS

Your groups are shown on your profile. [Change?](#)

**Houston**

A group all about Houston.

### 1.12.3 Update User

To quickly update a user's profile information, click (3) **Update User**. You'll be able to update their profile information on their behalf.

Connect Messages 4 Explore Admin Resources Christine G. ☰

#### UPDATE ALVIN F.

First name:

Last name:

Biography: 

Moved to NYC a few months ago, and lovin' every minute of it!

Timezone:

Facebook url:

Twitter handle:

Website url:

Default Notification Setting:  Send me an email for every new message

### 1.12.4 Update Permissions

To give a user additional Connect permissions, click (4) **Update Permissions**. From the permission update panel, you'll be able to give the user additional Connect privileges, such as the ability to edit other user's accounts, create groups, and add resources.

## UPDATE PERMISSIONS FOR ALVIN F.



accounts	user	Can change user
accounts	user	Can view banned users.
accounts	user	Can ban users.
accounts	user	Can unban users.
accounts	user	Can view user report.
accounts	user	Can view group report.
accounts	user	Can moderate all messages.
accounts	user	Can initiate direct messages.
accounts	invite	Can add invite
groups	group	Can add group
groups	group	Can change group
groups	group	Can edit any group.
media	image	Can promote an image
media	image	Can access the admin gallery
resources	resource	Can add resource
resources	resource	Can create resources anywhere.

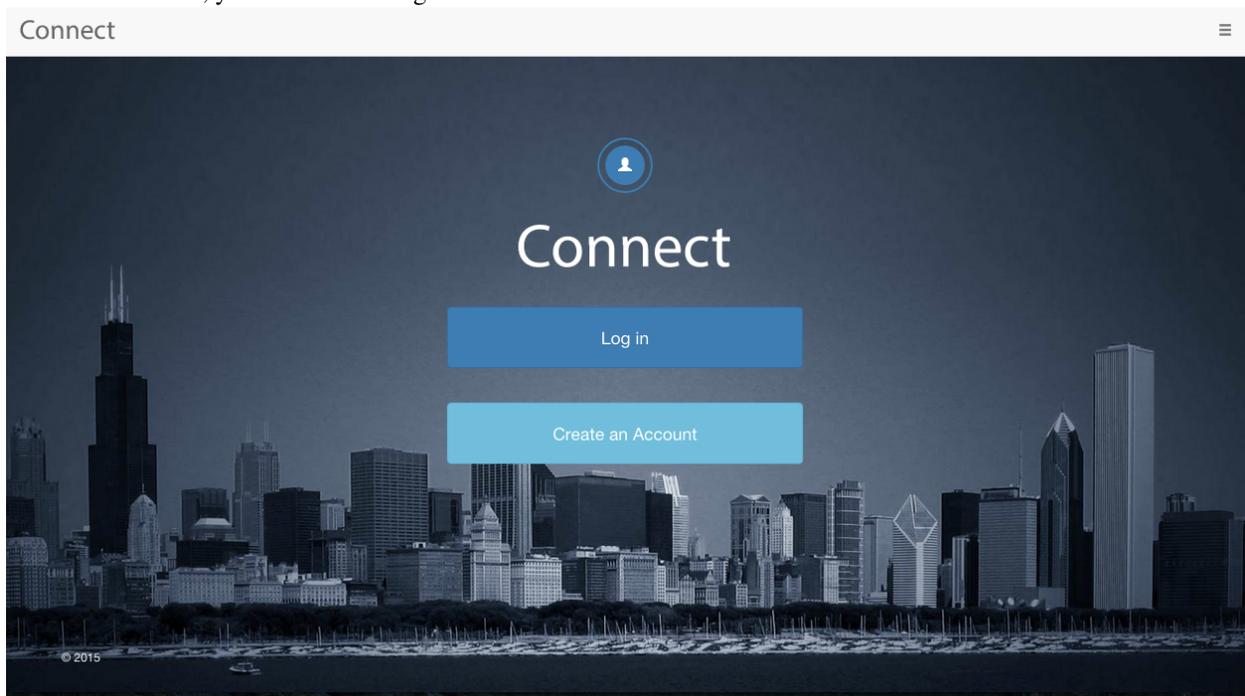
Save

This guide will cover basic features available to all Connect users, including:

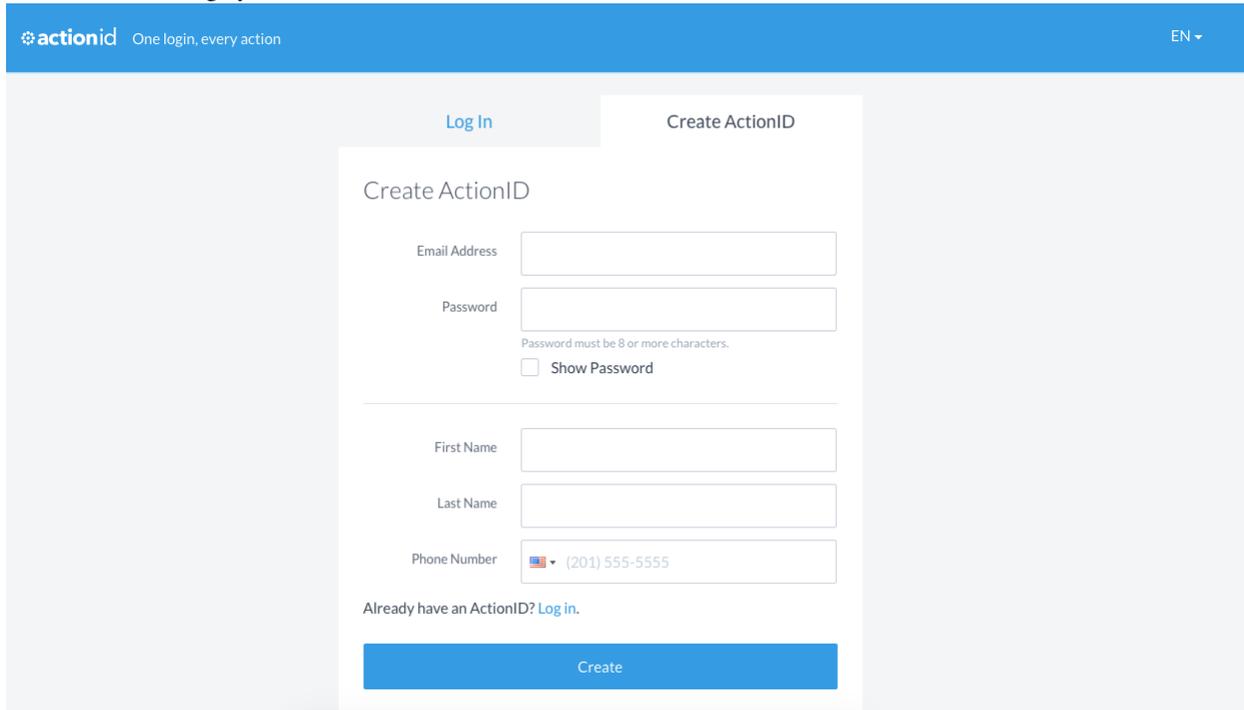
- Creating a Connect account
- Your Connect profile
- Connect groups
- Messages on Connect
- Managing your Connect inbox
- Accessing resources on Connect
- Managing your Connect subscriptions

## 2.1 Creating An Account

To access Connect, you will need to log in or create an account.



If you're creating an account, be sure to use the email address where you'd like to receive Connect messages and notifications. You will use this email address to log in to your Connect account moving forward. At this time, you will not be able to change your email address once it is set.



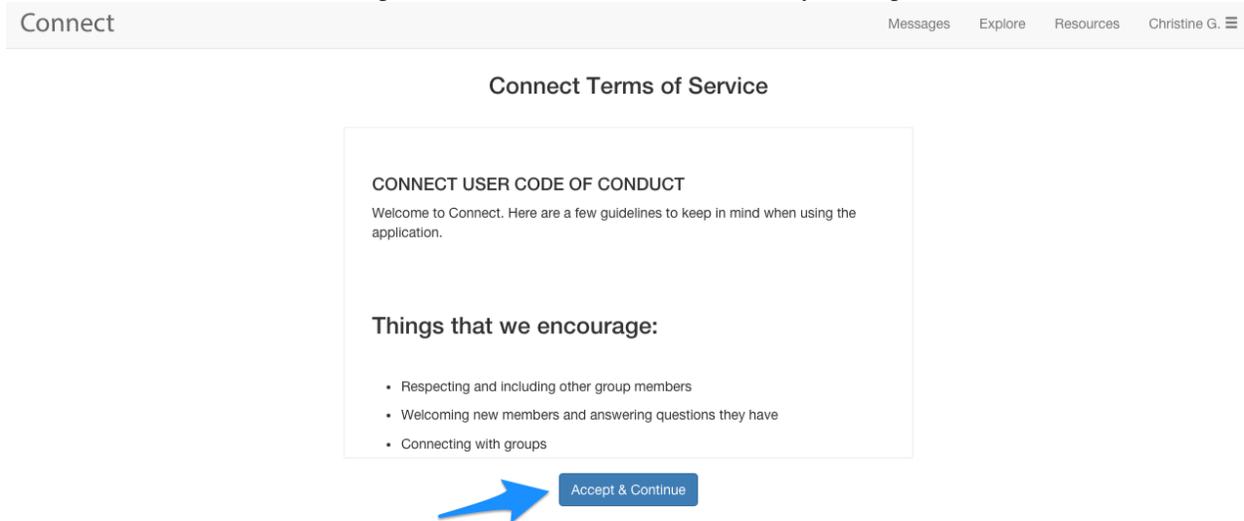
The screenshot shows the 'Create ActionID' form on the ActionID website. The header is blue with the ActionID logo and tagline 'One login, every action' on the left, and 'EN' with a dropdown arrow on the right. Below the header, there are two tabs: 'Log In' and 'Create ActionID'. The 'Create ActionID' form contains the following fields and elements:

- Email Address:** A text input field.
- Password:** A text input field with a note below it: 'Password must be 8 or more characters.' and a checkbox labeled 'Show Password'.
- First Name:** A text input field.
- Last Name:** A text input field.
- Phone Number:** A text input field with a country code dropdown set to '(201) 555-5555'.
- Already have an ActionID? [Log in.](#)**
- Create:** A blue button at the bottom of the form.

After your account has been created, you will be redirected to the Explore page on Connect. From here, you can find out which groups are available to join.

Using the navigation bar at the top of any screen on Connect, you can access this page again at any time by clicking **Explore**.

You will also need to read and accept the Connect Terms of Service before you can proceed within Connect.



The screenshot shows the 'Connect Terms of Service' page. The header is light gray with the word 'Connect' on the left and navigation links 'Messages', 'Explore', 'Resources', and 'Christine G.' with a menu icon on the right. The main content area is titled 'Connect Terms of Service' and contains the following text:

**CONNECT USER CODE OF CONDUCT**  
Welcome to Connect. Here are a few guidelines to keep in mind when using the application.

**Things that we encourage:**

- Respecting and including other group members
- Welcoming new members and answering questions they have
- Connecting with groups

At the bottom of the content area, there is a blue button labeled 'Accept & Continue' with a blue arrow pointing to it from the left.

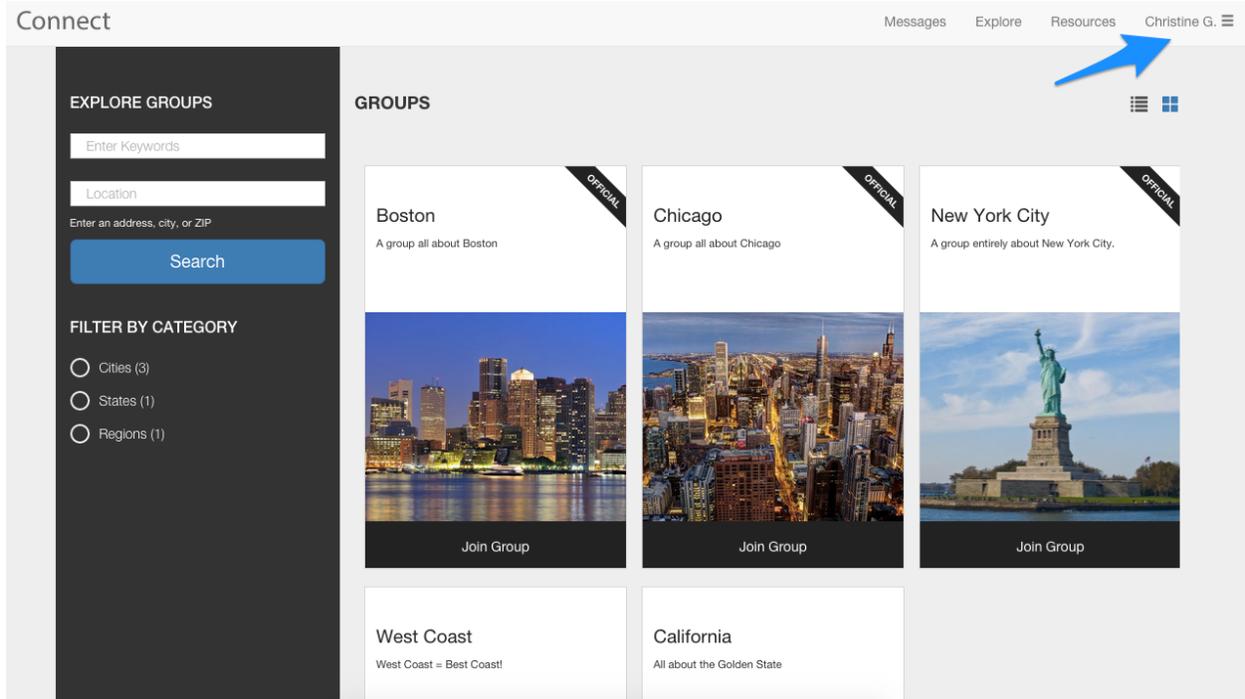
## 2.2 Your Connect Profile

Before joining groups, it's a good idea to create your Connect profile—that way, Connect group administrators will be able to find out more about you and what you're interested in.

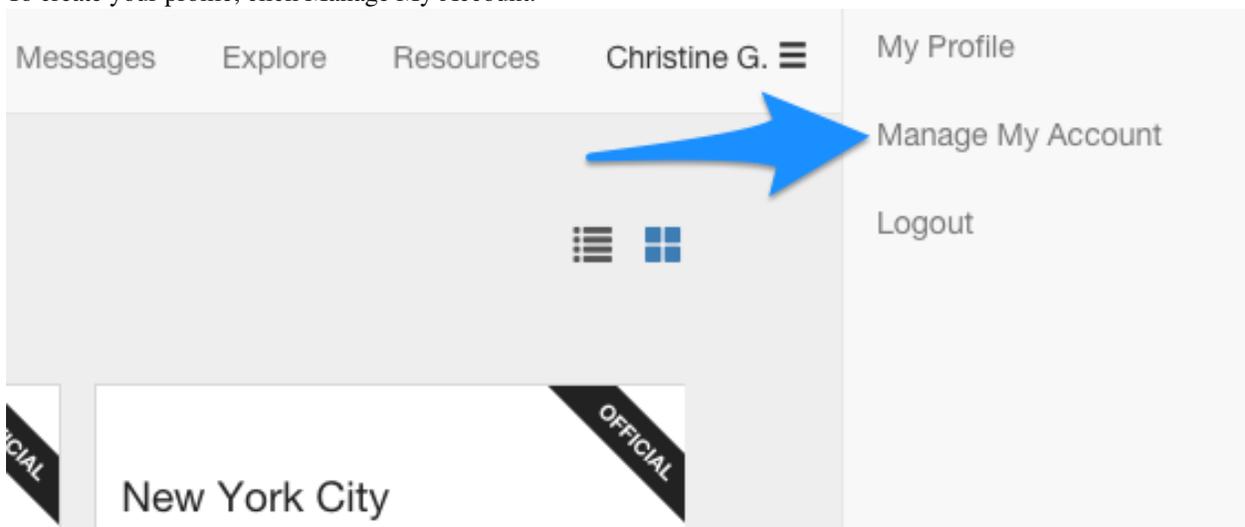
In this section, we'll cover (1) creating your profile and (2) viewing your profile and changing your profile picture.

### 1. Creating your profile

From the Connect navigation bar, click the three lines on the far right side. A menu will open displaying the following options: **My Profile**, **Manage My Account** and **Logout**.



To create your profile, click **Manage My Account**.



From the **Manage My Account** panel, you'll be able to:

Add your name—it will be displayed to other Connect users.

Fill out your biography. Share a little about yourself and why you're on Connect.

Indicate your current timezone. Make sure messages are displayed to you using the appropriate time.

Add your social media profile and website links.

Choose your default email notification settings. These settings determine how often you will receive Connect messages via email. (Find additional details on this below.)

Decide whether or not you'd like groups you're a member of displayed on your profile. If a group you're a member of has been set as private by the group administrator, only other members of that group will be able to see it on your profile.

Set a profile picture. This picture will be displayed on your profile, as well as next to your name in groups you're a member of. (Find additional details on this below.)

Connect Messages Explore Resources Christine G. ☰

---

UPDATE CHRISTINE G.

**a** First name   
Last name

Biography **b**

**c** Timezone

Facebook url

**d** Twitter handle

Website url

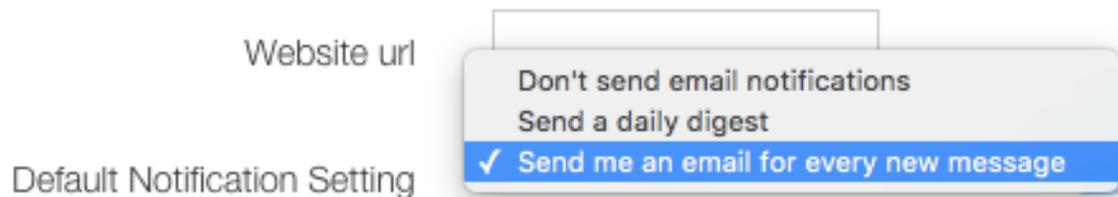
**e** Default Notification Setting

Show groups on profile  **f** Can we display the groups you belong to on your public profile?

**g** Image  No file chosen

To (e) update your default notifications, click the drop down menu and choose whether you'd like to receive an email for every new message you receive on Connect, receive a daily digest of messages, or receive no notifications.

After you've joined groups, you'll be able to further refine these settings on your profile.



To (g) add a profile picture, click **Choose File** and select the image you'd like to use for your profile picture.



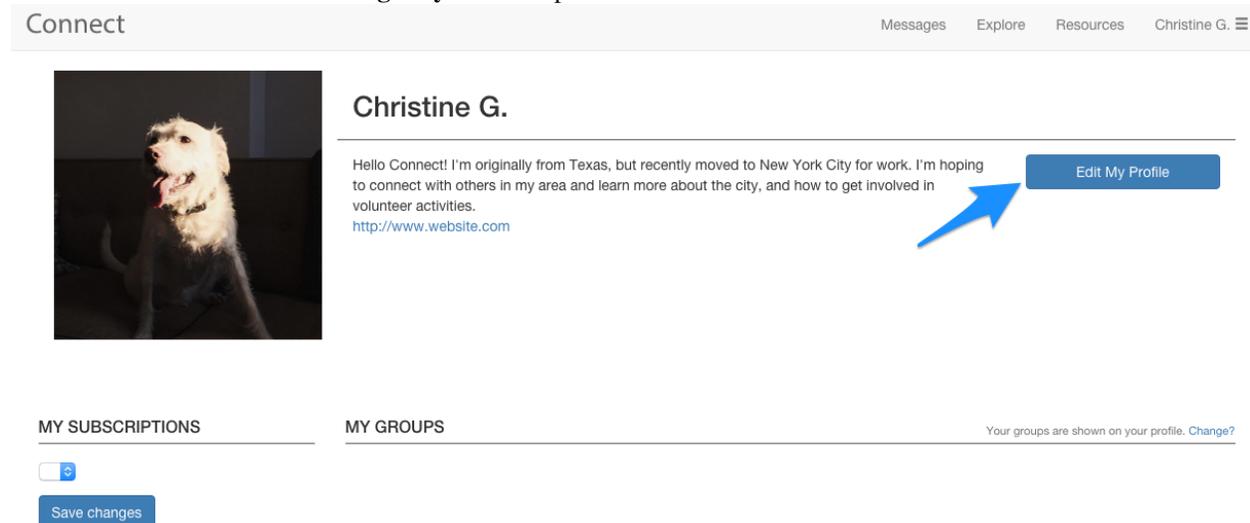
When you're done filling out your profile, click Update to save your changes.



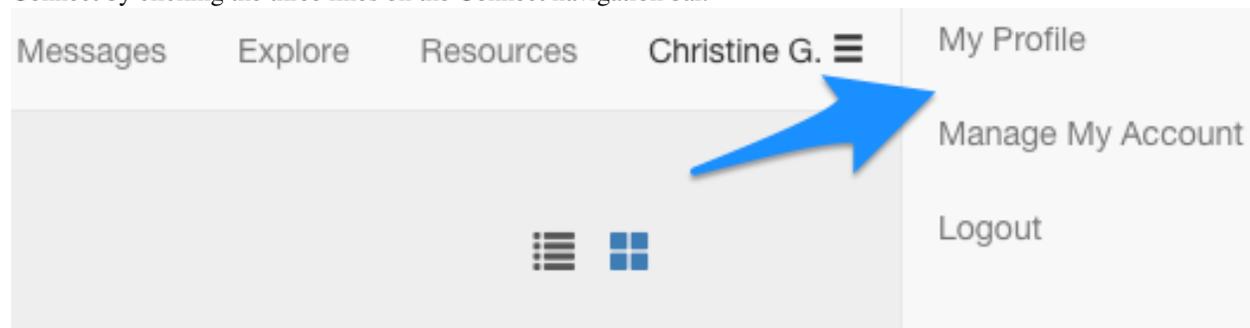
## 2. Viewing Your profile and changing your profile picture

After you click **Update** on the **Manage My Account** panel, you'll be redirected to your completed profile.

If you decide at this point to make additional updates to your profile, you can do that by clicking **Edit My Profile**. You will be sent back to the **Manage My Account** panel.



You can also access your profile (**My Profile**) or the profile editing feature (**Manage My Account**) from anywhere on Connect by clicking the three lines on the Connect navigation bar.



If you'd like to change your profile picture, you'll need to clear the original picture by checking the box marked **Clear**, then choose new profile picture and click **Update**.

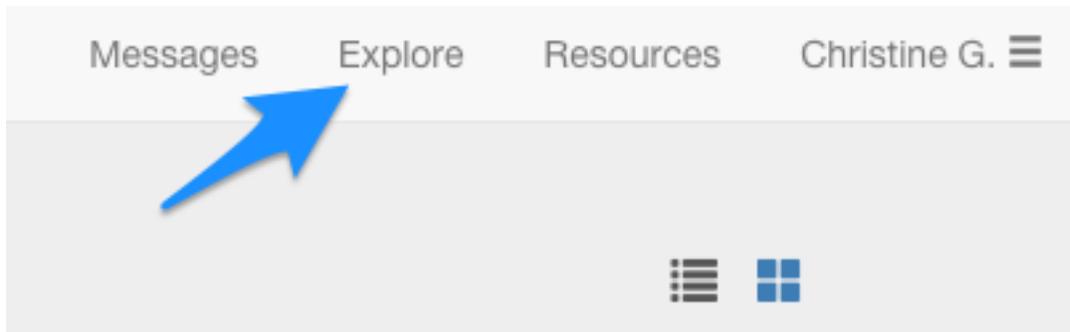


## 2.3 Connect Groups

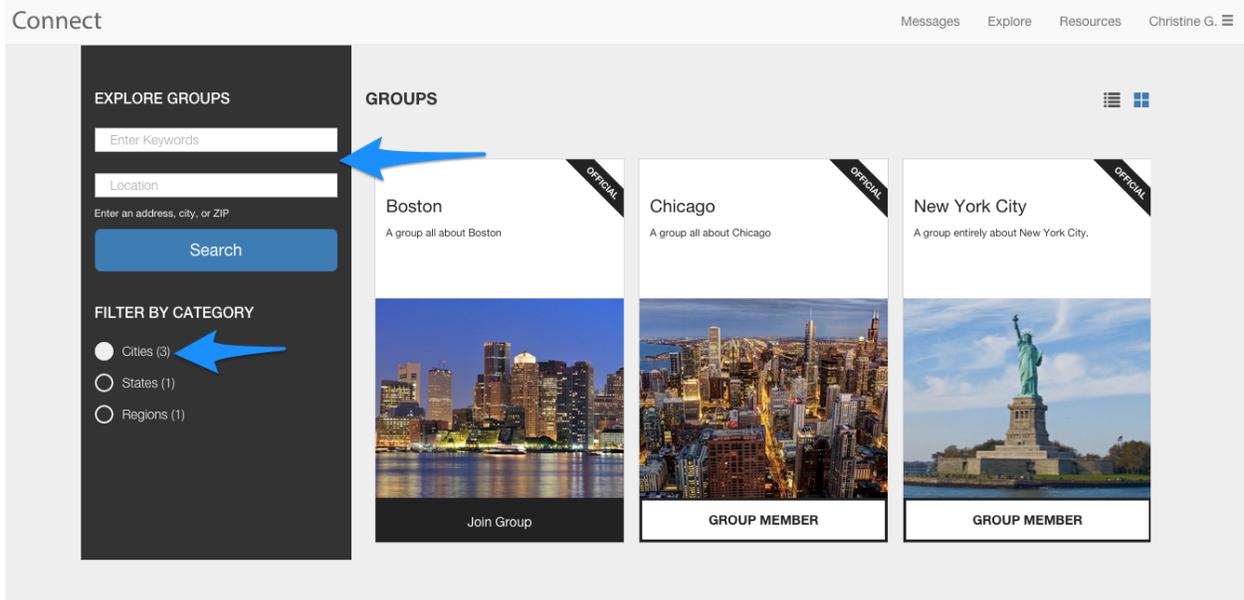
Now that your profile is complete, the next step in using Connect is joining groups. In this section, we'll cover:

1. Finding groups to join
  2. Joining groups
  3. Exploring groups
  4. Leaving groups.
1. Finding groups to join

To access a list of groups that are available for you to join, click **Explore** on the Connect navigation bar.

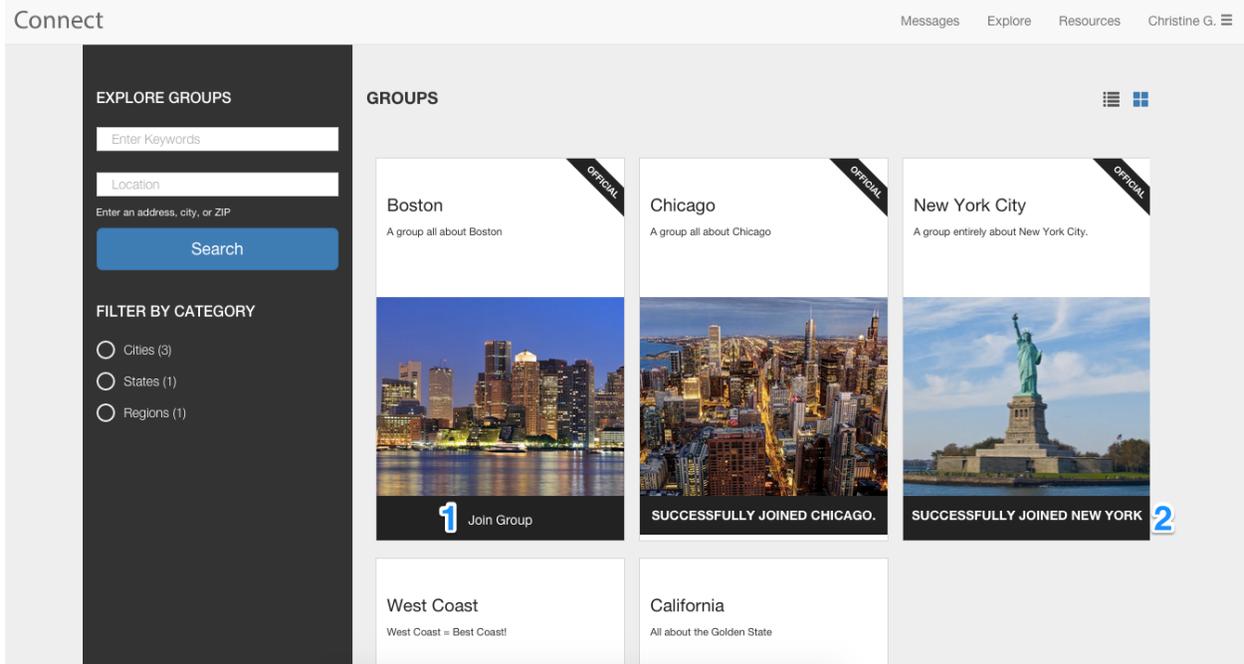


If you'd like to refine the list of groups being displayed, you can search for groups with keywords or by searching a location. You can also filter by group category—just click the name of the category you'd like displayed.



## 2. Joining groups

To join a group, click (1) **Join Group** on the group listing. You will receive (2) an instant confirmation indicating you have successfully joined the group.



You can also click the group name to preview the group, and will have the option to join the group from that page, as well.



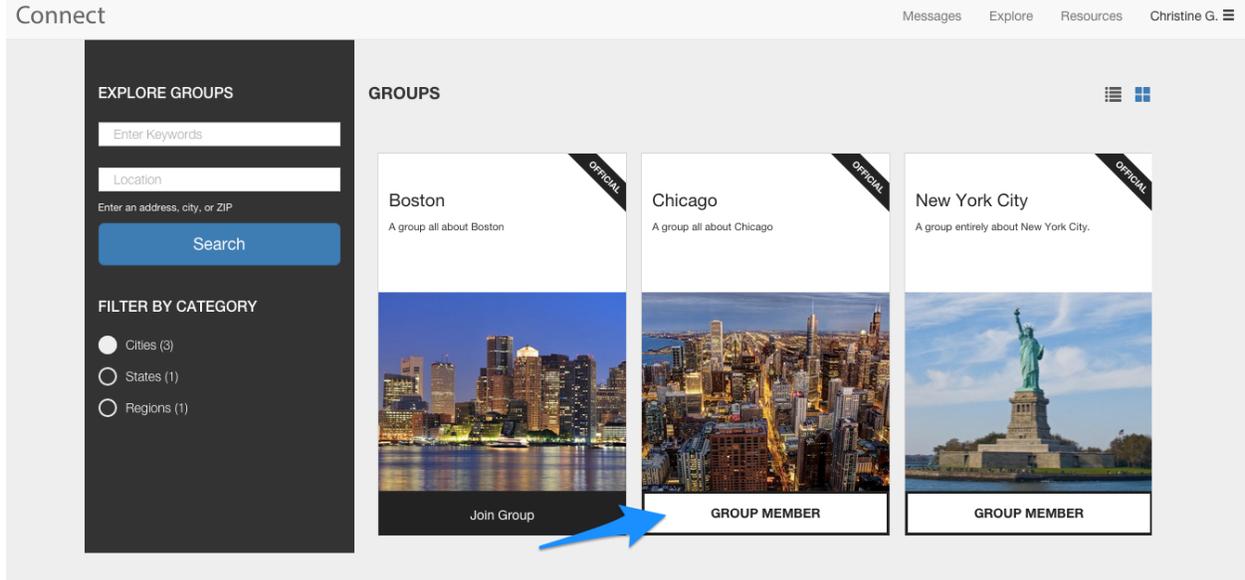
OWNERS: 2



DESCRIPTION

A group all about Boston

When you re-visit the Explore page, you will be able to see which groups you're currently a member of.



### 3. Exploring groups

Once you're a member of a group, you can find (a) more details about the group, (b) who is in your group, and (c) access resources that are available to members of your group.

If you'd like to (b) find out who else is a member of your group, click **See all members**.

The screenshot shows a Connect group page for 'New York City'. At the top right, there is a 'Messages' link. The group's profile picture is a circular image of the Statue of Liberty. Below the image, the group name 'New York City' is displayed, followed by 'Cities', 'Started: October 22, 2015', and 'New York, NY'. To the right of the profile picture are two buttons: 'MEMBER' and 'Leave'. Below the group information, there are four sections: 'OWNERS: 3' with three profile icons and a blue 'b' icon; 'MEMBERS: 7' with seven profile icons and a 'See all members' link; 'LOCATION' with a map showing the New York City area; 'DESCRIPTION' with the text 'A group entirely about New York City.'; 'RESOURCES' with a PDF icon and the title 'New York City subway map' and a 'See all resources' link; and 'LATEST THREADS' with a 'Send a Message' button and a thread snippet starting with 'Welcome new members!'.

Connect staff members will be designated with a banner that says "Staff Member". To find out more information about a group member, click their name to view their profile.

If the group member is also a group owner, you can send them a direct message by clicking the envelope icon to the right of their name. Find more details on messaging group owners in the section of this guide titled Messages on Connect.

NEW YORK CITY

4 members

GROUP OWNERS

Brock D.

Christine G.

William G.

GROUP MEMBERS

Audrey R.

Jay F.

Nick C. STAFF MEMBER

Shai S.

To (c) access group resources, click the name of the resource to view it in a new browser tab. Or, click &ldquo;See all resources&rdquo; to access the Resources page for the group. Find more details on resources in the section of this guide titled Accessing Resources on Connect.

## RESOURCES



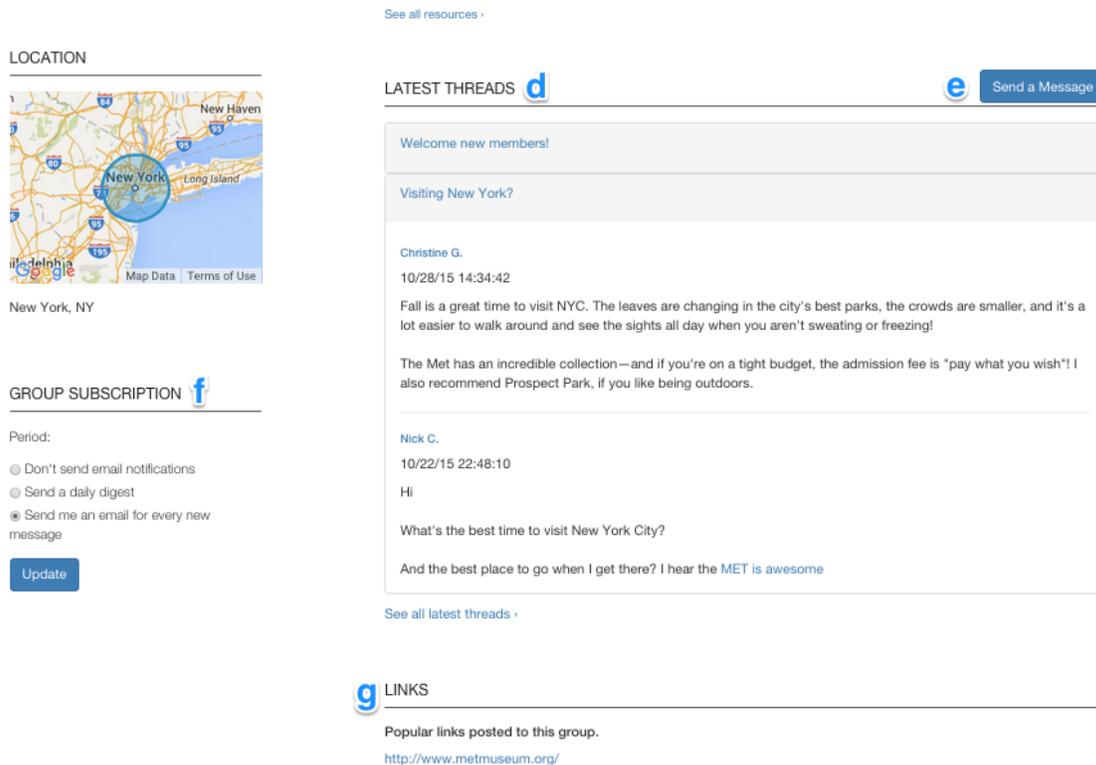
New York City subway map

[See all resources >](#)





Further down the group page, you can also (d) read the latest threads in the group, (e) send a message in the group (Find more details on sending messages in the section of this guide titled Messages on Connect.), update your (f) group subscription notifications, and (g) find the most popular links posted to the group.



The (d) latest threads view will have all but the most recent threads collapsed. To expand the thread view, click the subject line of a collapsed message.

[Welcome new members!](#) 

[Audrey R.](#)  
11/14/15 22:32:48  
Thanks for the welcome! The subway map was very helpful.

[Visiting New York?](#)

[Christine G.](#)  
10/28/15 14:34:42  
Fall is a great time to visit NYC. The leaves are changing in the city's best parks, the crowds are smaller, and it's a lot easier to walk around and see the sights all day when you aren't sweating or freezing!  
  
The Met has an incredible collection—and if you're on a tight budget, the admission fee is "pay what you wish"! I also recommend Prospect Park, if you like being outdoors.

---

[Nick C.](#)  
10/22/15 22:48:10  
Hi  
  
What's the best time to visit New York City?  
  
And the best place to go when I get there? I hear the [MET is awesome](#)

To (f) update your notifications for this group, choose your desired message frequency. Click **Update** to save your changes.

## GROUP SUBSCRIPTION

---

Period:

- Don't send email notifications
- Send a daily digest
- Send me an email for every new message



You'll see a confirmation message at the top of the group page indicating your change has been made. Find more details on managing your group subscriptions in the section of this guide titled Managing Connect Subscriptions.

Your subscription preference has been updated. x

### 4. Leaving groups

If you've decided you'd no longer like to be a member of a group, there are a few ways you can leave the group.

From the group page, you can end your group membership by clicking the **Leave** button.

Connect Messages Explore Resources Christine G. ☰



MEMBER

Leave

**New York City**

Cities

Started: October 22, 2015

New York, NY

OWNERS: 3



DESCRIPTION

A group entirely about New York City.

You can also leave groups from your profile by clicking **Leave Group** on the group listing.

Connect Messages Explore Resources Christine G. ☰

**Christine G.**

Hello Connect! I'm originally from Texas, but recently moved to New York City for work. I'm hoping to connect with others in my area and learn more about the city, and how to get involved in volunteer activities.  
<http://www.website.com> Edit My Profile

**MY SUBSCRIPTIONS**

Chicago

Chicago

- Don't send email notifications
- Send a daily digest
- Send me an email for every new message

Save changes

**MY GROUPS** Your groups are shown on your profile. [Change?](#)

Chicago official

A group all about Chicago

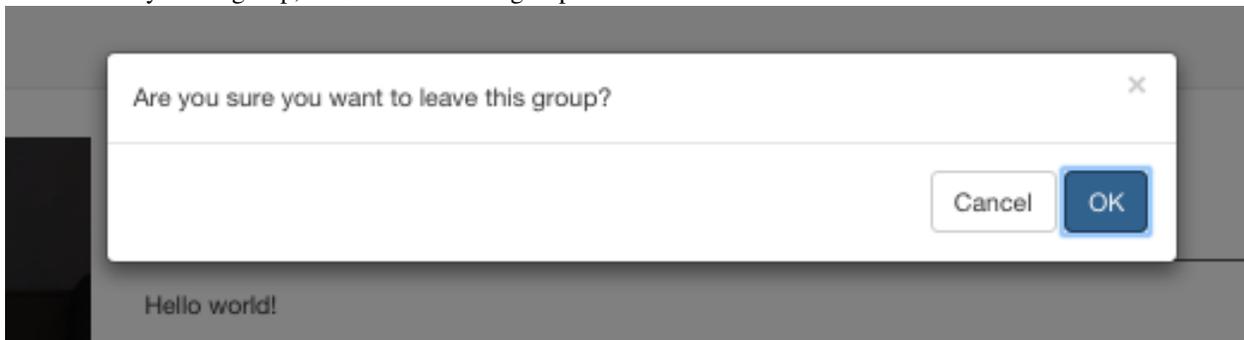
Leave Group

New York City official

A group entirely about New York City.

Leave Group

In both cases, you will receive a confirmation dialog box asking if you are sure you'd like to leave the group. Click **Cancel** to stay in the group, or **OK** to leave the group.



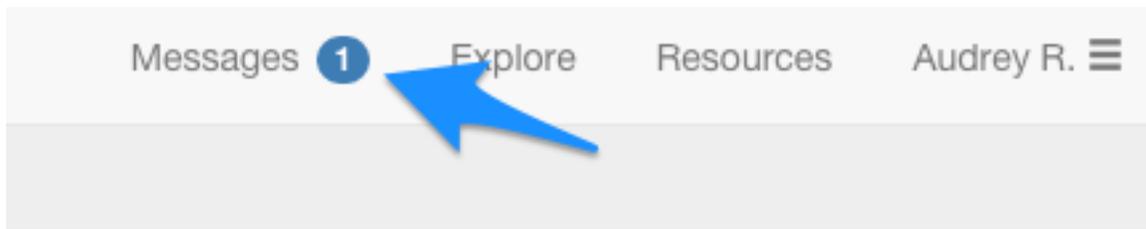
## 2.4 Messages On Connect

Once you become a member of a group on Connect, you'll be able to send and receive group messages. In this section, we'll cover how to:

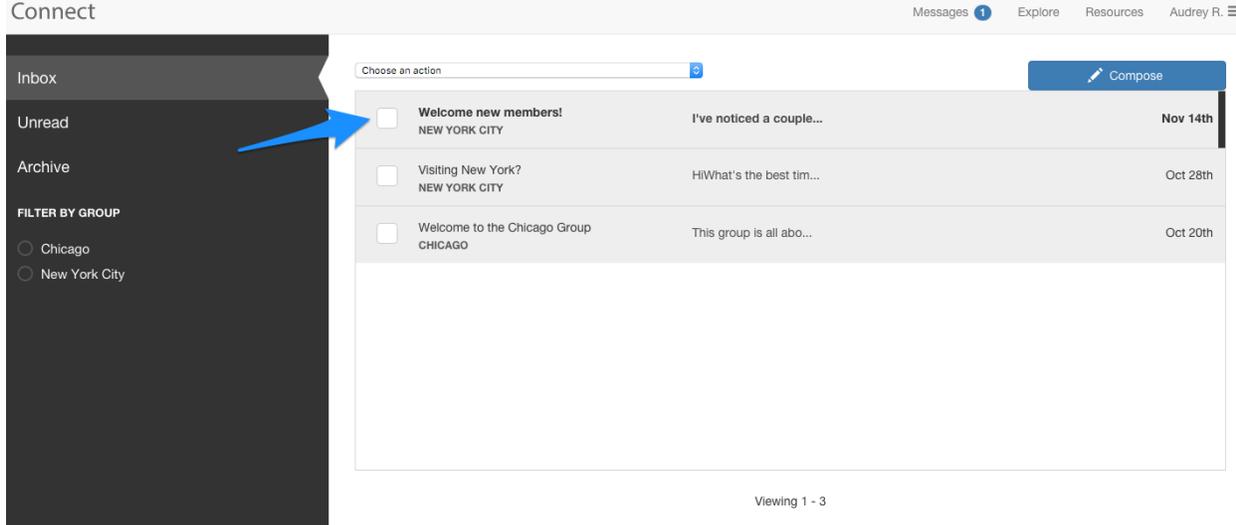
- Find your new Connect messages
- Reply to messages and compose new messages
- Send messages to Connect administrators
- Flag messages as inappropriate

### 1. Find and read new messages

When you receive new Connect messages, a badge indicating the number of unread messages you have will appear on the navigation bar. Click **Messages** to go to your inbox.

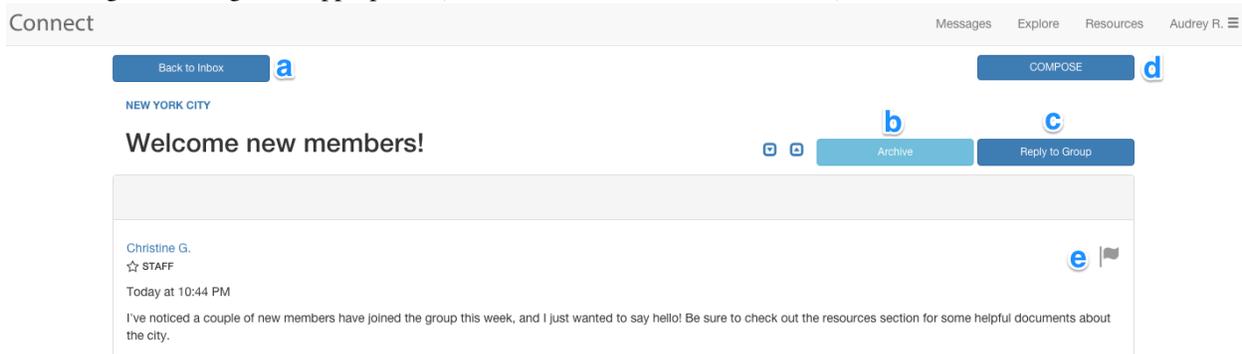


In your Connect inbox, unread messages will appear in bold. To read the full message, click anywhere on the message.



From the message, can read the full text and any replies in the message thread. You can also take several other actions including:

1. Return to the inbox
2. Archive the message
3. Reply to the message thread (Find additional details on this below.)
4. Compose a new message (Find additional details on this below.)
5. Flag the message as inappropriate (Find additional details on this below.)



To (a) return to your inbox from the message, simply click the Back to Inbox button. The message will now be marked as read, but will remain in your inbox.

If you'd like to mark the message as read, and archive it so it no longer appears in your inbox, click (b) **Archive**.



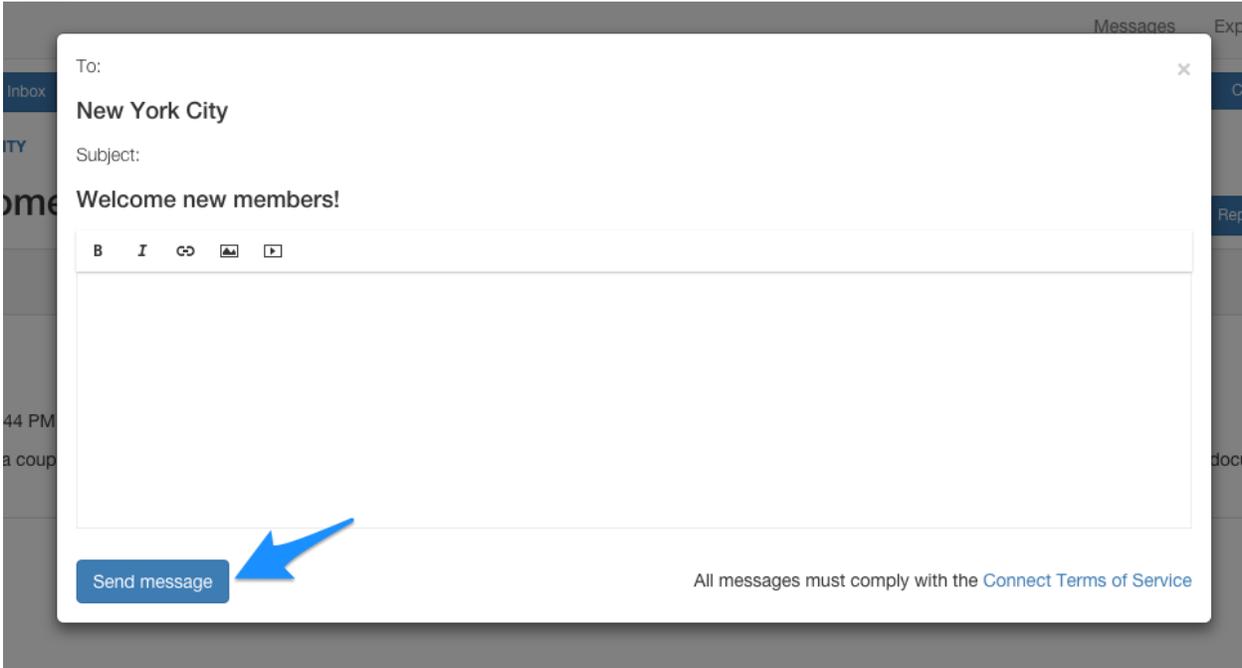
2. Reply to messages and compose new messages

If you'd like to reply to the message, click (c) **Reply to Group**. Compose your message, and click **Send Message**. Your reply will be seen by all members of the group.

Click (d) **Compose** to start a new message thread in a group you're the member of.

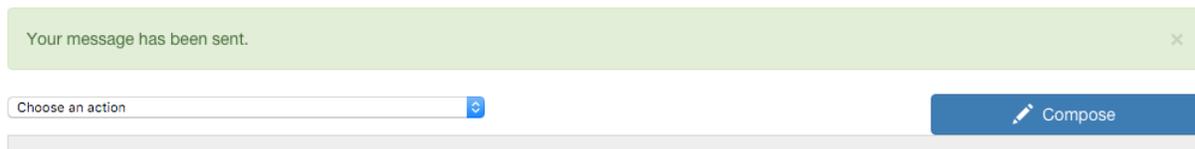


When you (c) send a reply to the message thread, the group name and subject line will be indicated in the compose window.



When you (d) start a new message thread, you'll need to choose the group you want to send the message to and compose a subject line. Click **Send Message** to send your message.

Once you click **Send message**, you will be redirected to your Connect inbox and will be given a confirmation message that it was sent. The sent message will be viewable from your inbox.



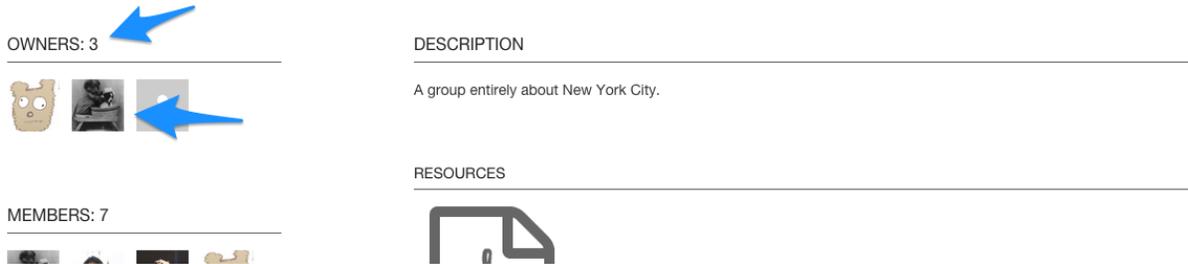
Click on the message to view what you sent, or any replies to it.

Please note: If the group you are a member of is set as moderated, a group administrator will need to approve your message before it will be viewable by other group members.

### 3. Send messages to Connect administrators

You can send the owners (or administrators) of your Connect groups direct messages from two places: from the member listing in a group page, or from the administrator's Connect profile. You can access both from the Connect group.

Click **Owners** to send the message from the member listing, or click the administrator's profile picture icon to go to their profile.

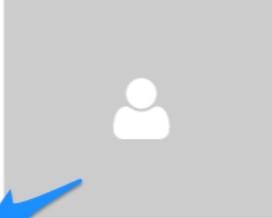


If you choose to click **Owners**, you'll be sent to the group member panel. Find the administrator (owner) you'd like to message and click the envelope icon to the right of their name.

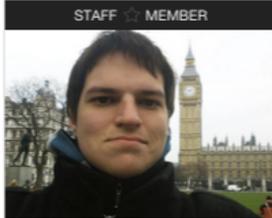
NEW YORK CITY

## 4 members

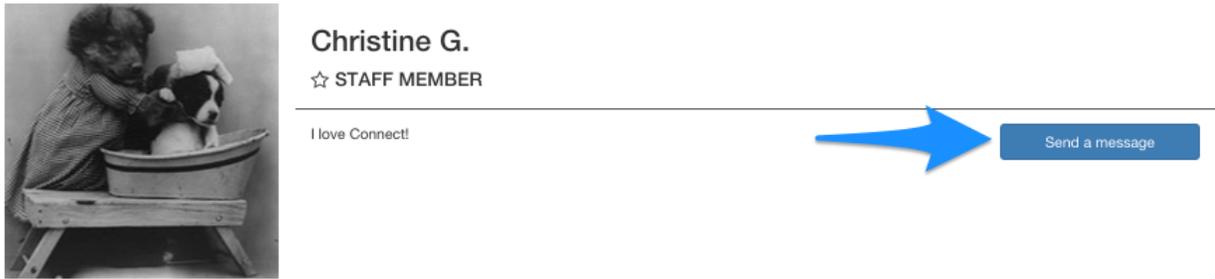
### GROUP OWNERS

 Brock D. 	 Christine G. 	 William G. 
---	---	--

### GROUP MEMBERS

 Audrey R. 	 Jay F. 	 Nick C. 	 Shai S. 
--	---	---	--

If you choose to click their profile picture icon, you will be sent to the administrator's profile. From here, click the **Send a message** button.

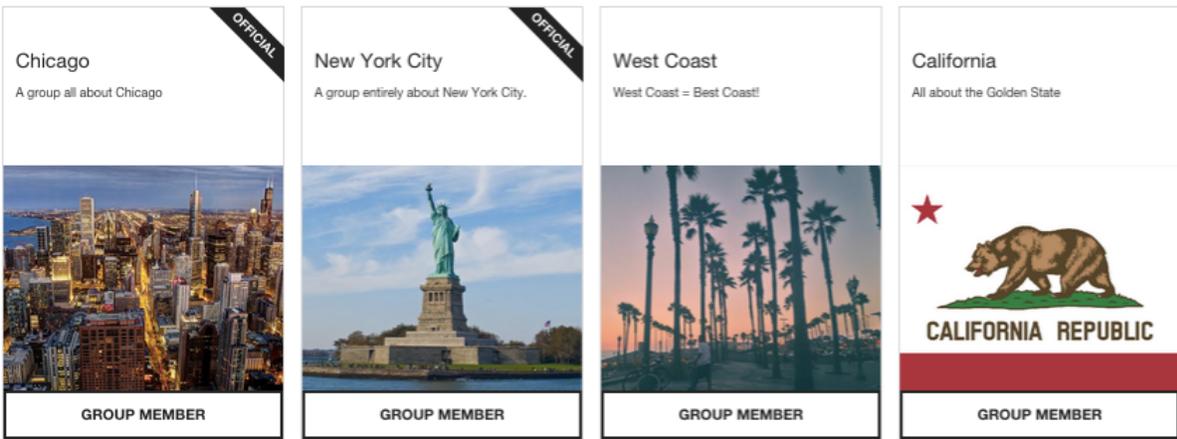


**Christine G.**  
☆ STAFF MEMBER

I love Connect!

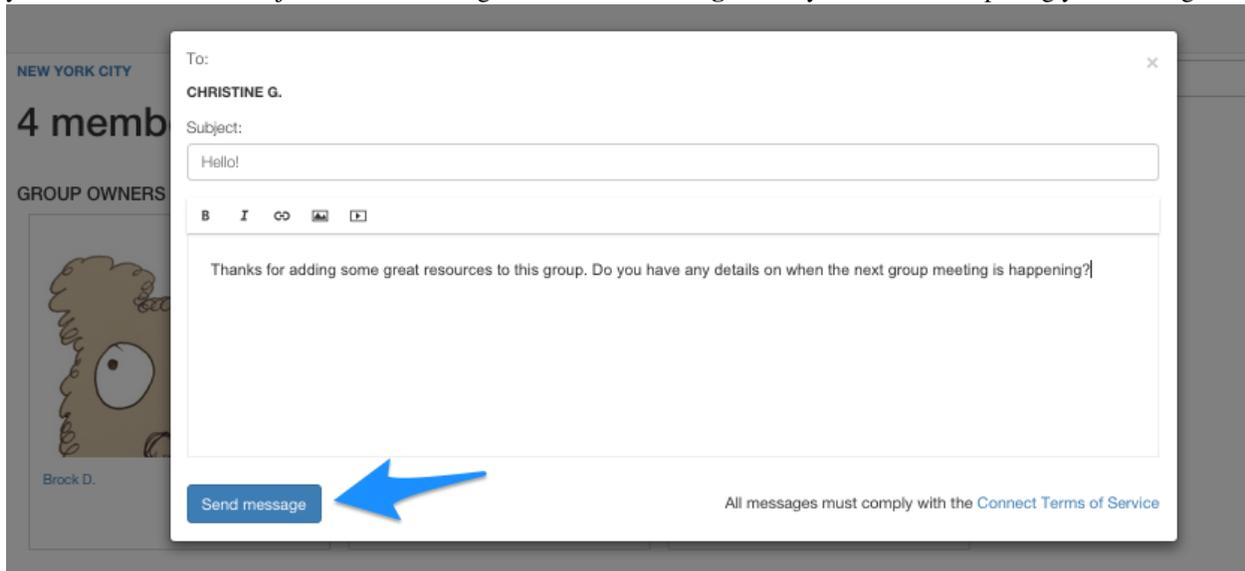
Send a message

CHRISTINE G.'S GROUPS



<b>Chicago</b> A group all about Chicago  GROUP MEMBER	<b>New York City</b> A group entirely about New York City.  GROUP MEMBER	<b>West Coast</b> West Coast = Best Coast!  GROUP MEMBER	<b>California</b> All about the Golden State  GROUP MEMBER
---	---	--	---

In both cases, after clicking, you will see a compose window. The administrator's name will be pre-filled, but you'll need to fill in a subject line and message. Click **Send message** when you're done composing your message.



NEW YORK CITY  
4 memb  
GROUP OWNERS  
Brock D.

To: CHRISTINE G.

Subject: Hello!

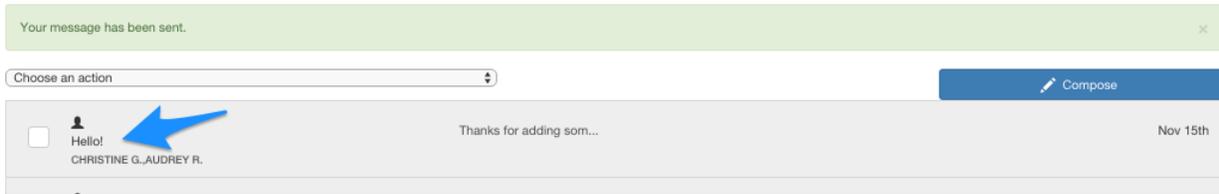
B I   

Thanks for adding some great resources to this group. Do you have any details on when the next group meeting is happening?]

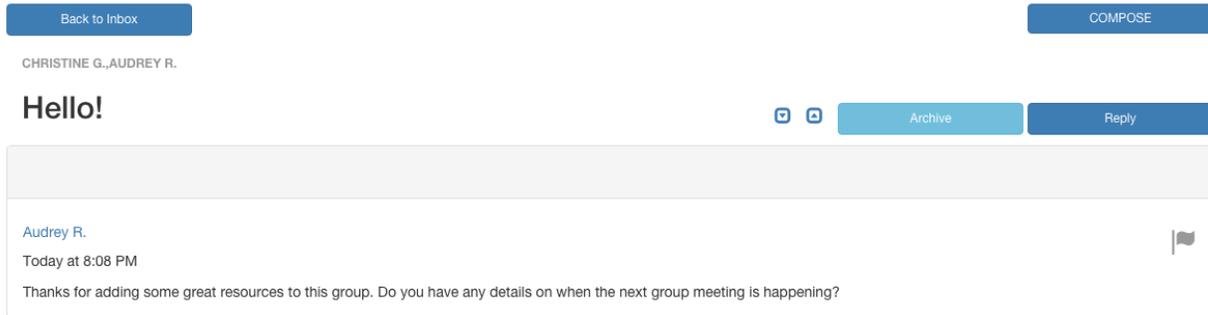
Send message

All messages must comply with the Connect Terms of Service

Once you click **Send message**, you will be redirected to your Connect inbox and will be given a confirmation message that it was sent. The sent message will be viewable from your inbox.



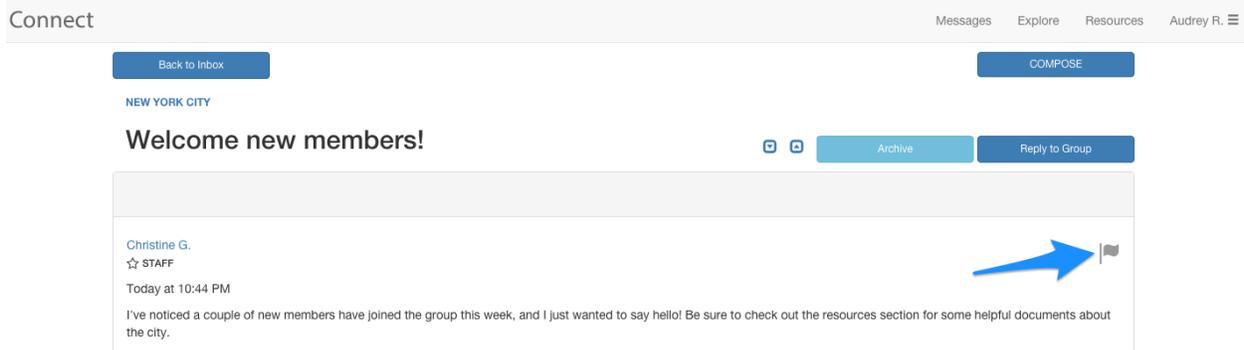
As with other messages, click on the message to view what you sent or any replies to it.



#### 4. Flagging messages

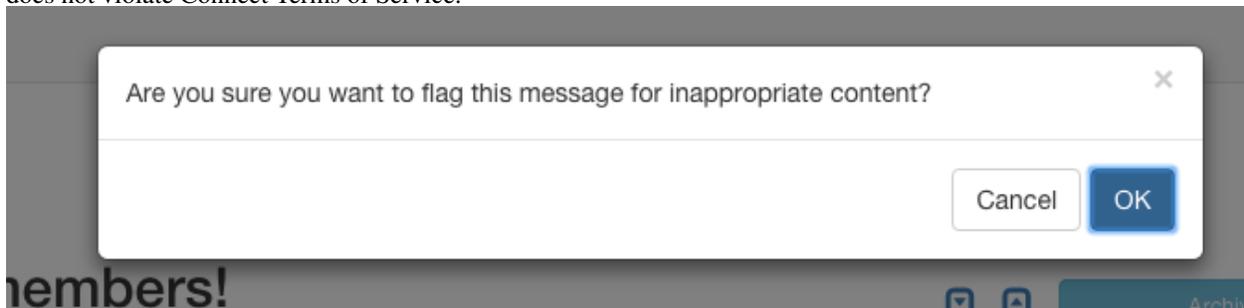
If you receive a message in your inbox that you think is in violation of the Connect Terms of Service, you can (e) flag the message as inappropriate and it will be reviewed by a Connect group administrator.

Click the flag icon in the body of the message.



You will receive a message confirming if you would like to flag the message as inappropriate.

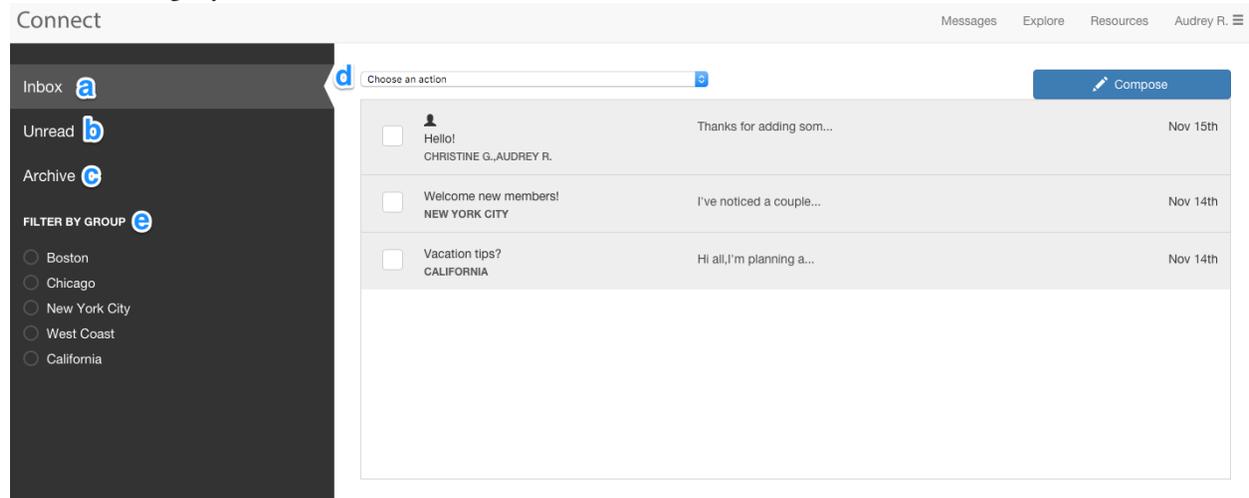
Flagged messages will no longer appear in your Connect inbox—unless a group administrator determines the message does not violate Connect Terms of Service.



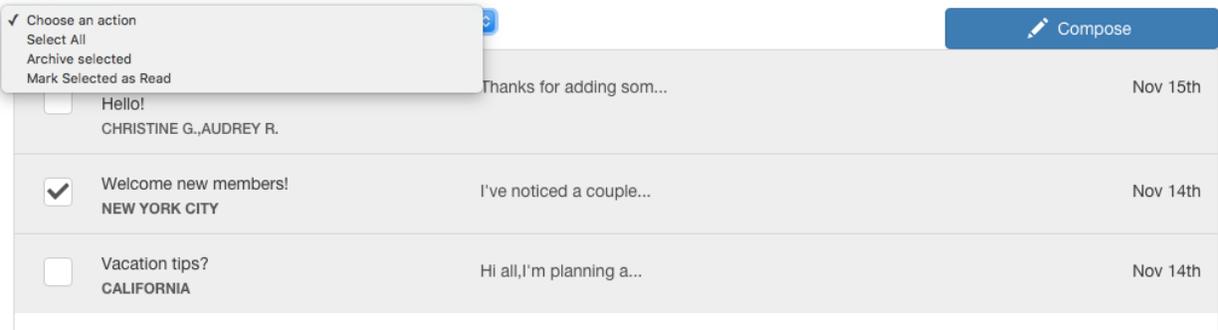
## 2.5 Managing Your Connect Inbox

Your Connect inbox will contain messages from every group you're a member of, as well as direct messages between you and Connect administrators. To make it as simple as possible to find relevant messages, there are filtering features available on your inbox.

The default view, (a) **Inbox**, will display your entire inbox—this includes any new messages or read messages you have not archived. The (b) **Unread** view contains only new messages you haven't read yet. The (c) **Archive** view contains messages you have marked to be archived.

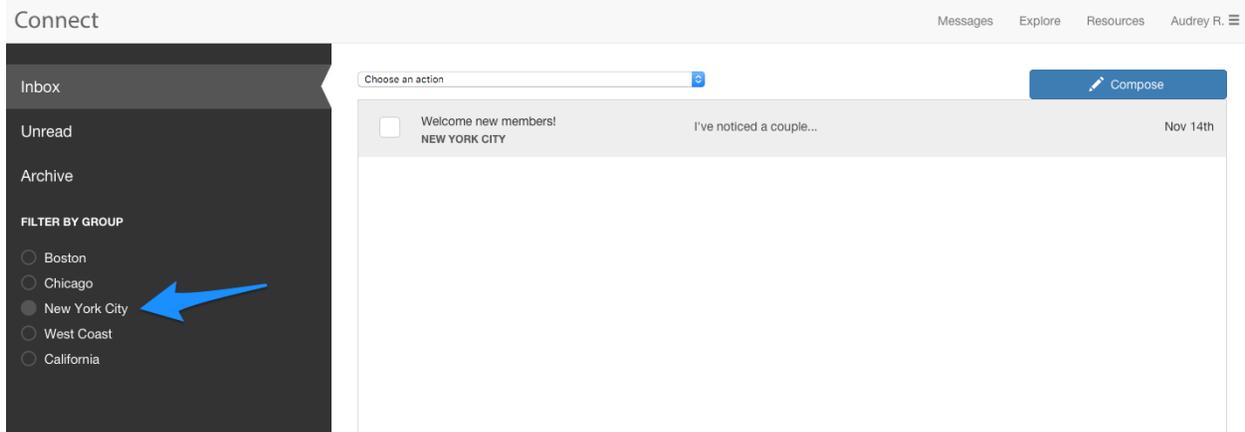


By using the (b) **Choose an action** feature, you can manage your messages in bulk. Select all messages in your inbox, archive all selected messages, or mark the selected messages as read.



If you're a member of several Connect groups, and would like to just see messages from one group in particular, you can do that by utilizing the (e) **Filter By Group** feature.

Click the circle next to the group name and choose whether you'd like to view the Inbox, Unread or Archive message views.



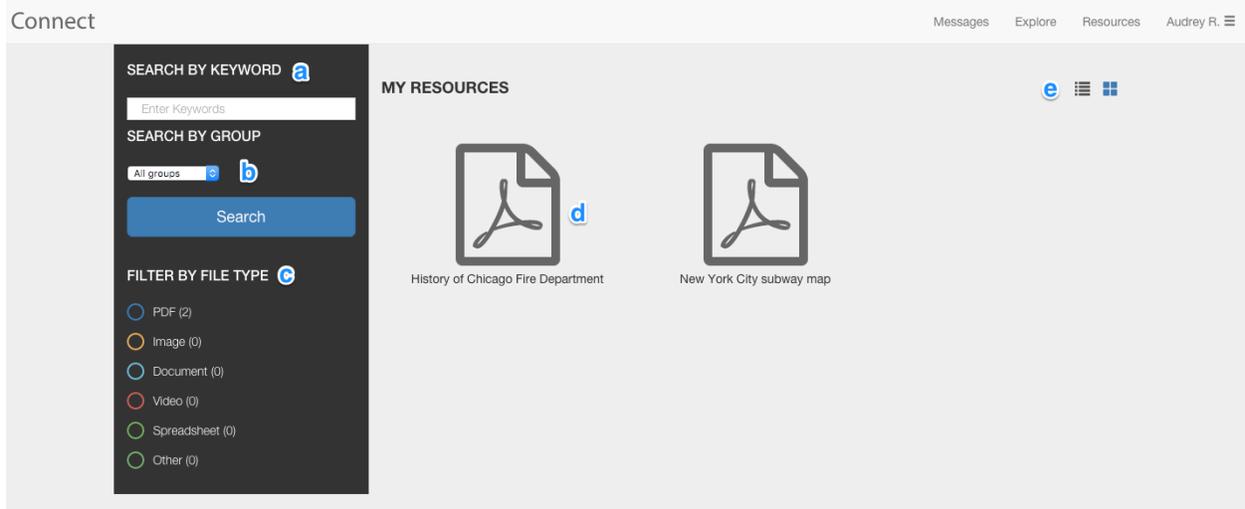
## 2.6 Accessing Resources On Connect

There are two ways to access resources on Connect. You can access all resources shared with groups you are a member of in the Resources panel. Or, you can access resources in individual groups by clicking select resources from the group's page.

To access the Resources panel, click **Resources** on the Connect navigation bar.



From the Resources panel, you can search for resources by (a) keyword, (b) or by group. You can also (c) filter the resources by file type. Click the file icons under **My Resources** to access them immediately. Depending on the file type, they will either open in a new browser tab, or begin downloading to your computer.



If you prefer you can also (e) change the resource view from grid to list view.

**SEARCH BY KEYWORD**  
Enter Keywords  
**SEARCH BY GROUP**  
All groups  
Search

**FILTER BY FILE TYPE**

- PDF (2)
- Image (0)
- Document (0)
- Video (0)
- Spreadsheet (0)
- Other (0)

**MY RESOURCES**

- History of Chicago Fire Department
- New York City subway map

To access resources in groups, click the file icon in the group view to access the file immediately.

MEMBER  
Leave

**New York City**  
Cities  
Started: October 22, 2015  
New York, NY

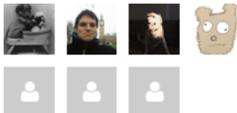
### OWNERS: 3



### DESCRIPTION

A group entirely about New York City.

### MEMBERS: 7



[See all members](#)

### RESOURCES



New York City subway map

[See all resources](#)

Or click **See all resources** to open the group resources page.

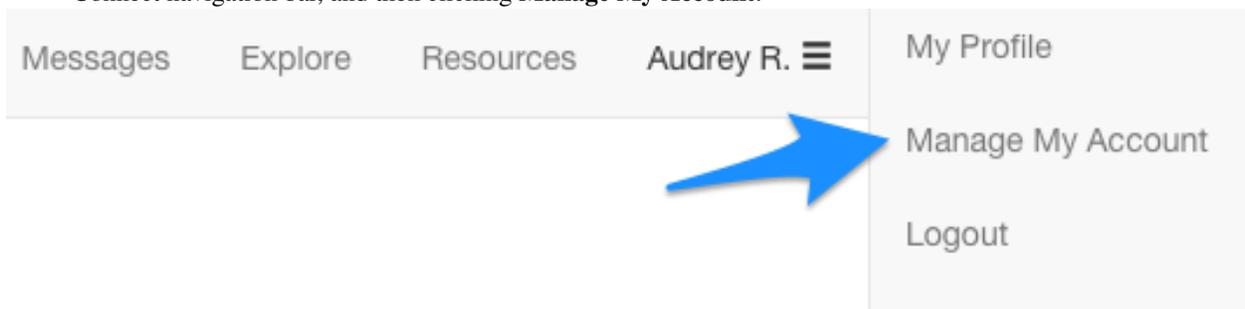


## 2.7 Managing Your Connect Subscriptions

When you're a member of Connect groups, you will receive messages from the group in your Connect inbox and at the email address you used to create your Connect account.

There are three ways you can make changes to your Connect subscription notifications: (1) From the **Manage My Account** panel, (2) from **My Profile**, and (3) from individual group pages.

1. Edit your subscription notifications from the **Manage My Account** panel by clicking the three lines on the Connect navigation bar, and then clicking **Manage My Account**.



From **Manage My Account**, you can choose default notification settings for all of your subscriptions. Click **Update** to save your changes.

Please note: If you choose to receive no email notifications, or a daily digest, this will apply to all of your Connect messages.

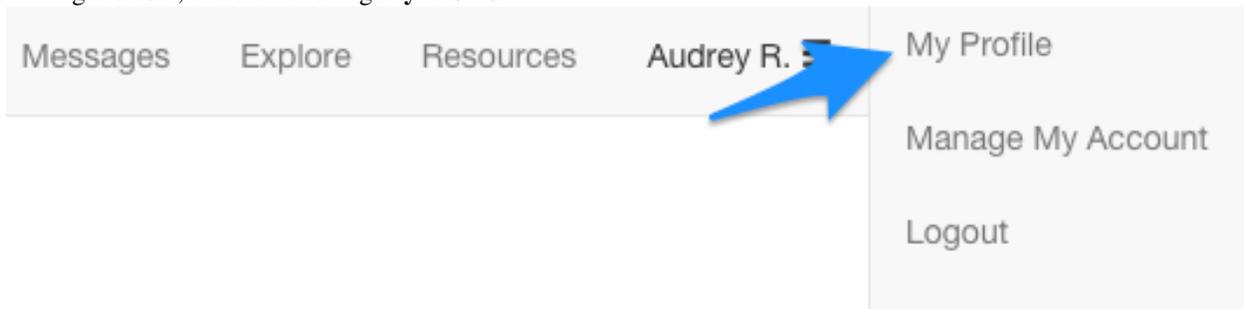
Twitter handle

Website url

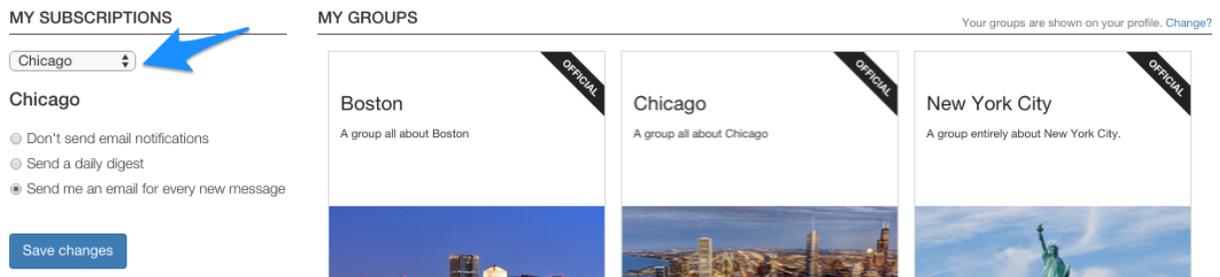
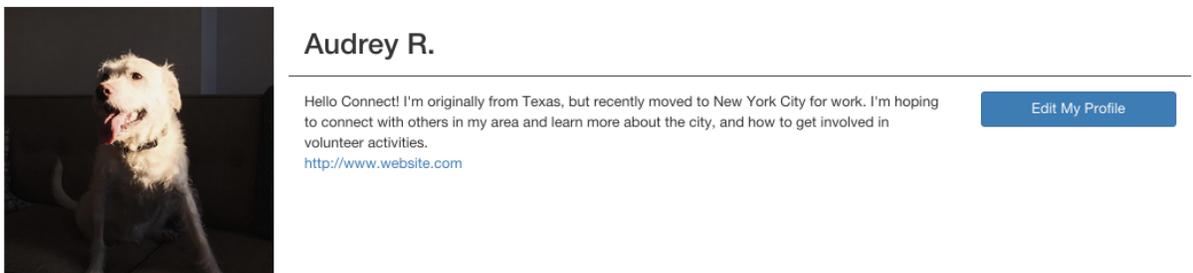
Default Notification Setting

- Don't send email notifications
- Send a daily digest
- Send me an email for every new message

2. Edit your subscription notifications from the My Profile panel by clicking the three lines on the Connect navigation bar, and then clicking **My Profile**.



From your profile, under **My Subscriptions**, choose the group you would like to manage your notifications for. Choose from no email notifications, a daily digest, or an email for every new message. Click **Save changes** to update your subscription settings.



2. Edit your subscription notifications from individual groups by going to the group page and choosing the notification settings.

Choose from no email notifications, a daily digest, or an email for every new message. Click **Update** to save your changes.

## GROUP SUBSCRIPTION

---

Period:

- Don't send email notifications
- Send a daily digest
- Send me an email for every new message

Update



---

## Developer and Operator Guide

---

Connect is written in Python using the [Django](#) library.

### 3.1 Running Connect

The easiest way to run Connect is by deploying it as a [Heroku](#) app with static files and uploaded content stored on [Amazon S3](#). However, support for [Docker](#) is also available, which is useful for doing custom deployments into existing environments as well as for local development.

#### 3.1.1 Guides

##### Deploying Connect to Heroku

The easiest way to test out Connect is to deploy it via [Heroku](#) backed by [Amazon S3](#) for storage and (if possible) [Amazon Simple Email Service \(SES\)](#) for outgoing email.

##### Prerequisites

There are 4 things you need to start with:

1. A verified Heroku Account, with the Heroku Toolbelt installed locally
2. An Amazon Web Services Key/Secret
3. A S3 Bucket the above Key/Secret has access to. This will store static assets and uploads.
4. A Simple Email Service verified address you're willing to use for outgoing Connect email

**Heroku** This tutorial assumes that you have a [verified Heroku account](#) with an attached credit card and the [Heroku Toolbelt](#) installed locally on your machine and available to you via command line.

**Amazon** It also assumes that you have an Amazon Web Service Key and Secret, which has full access to S3 and (if you want to use it) the SES email address or domain you wish to send email from.

**Static Asset & Upload Storage** Connect uses Amazon S3 by default to store both static assets and uploaded files. You'll need a S3 bucket which Connect can use to store these assets. You can define the specific folder you want any one installation of Connect to store its files in, but this tutorial will put uploaded files in the `connectdemo/uploads` folder and static assets in the `connectdemo/staticfiles` folder.

In order for fonts to correctly work on Connect, you'll need to enable Cross-Origin Resource Sharing on the bucket you'll be storing static assets on. More details on how to do this can be found at [Hosting Connect Fonts on Amazon S3](#)

**Outgoing Email via Simple Email Service** If you want to use SES, the same Key/Secret you use for Amazon will need to have outgoing email rights on SES and the email address you want notifications to be sent from needs to be a [verified outgoing email address](#) or part of a [verified domain](#).

More details about setting up outgoing email can be found at [Outgoing Email Configuration](#).

### Deploying Connect to Heroku

The initial deployment of Connect involves these steps:

1. Clone the Connect repo
2. Create a Heroku App
3. Add Add-ons for the Heroku App
4. Configure Core Connect Settings
5. Configure S3 Static & Upload Storage
6. Deploy Connect to Heroku
7. Setup the Connect Database
8. Deploy Static Assets
9. Scale up Connect

#### Clone the Connect repo

```
# Clone the Connect Repository
git clone https://github.com/ofa/connect.git

# Switch to the new local Connect folder
cd connect
```

#### Create the Heroku App

```
# Replace 'public-connect' in all these commands with the app name you want
heroku apps:create public-connect

# Use the Heroku Multi-Buildpack
heroku buildpacks:set https://github.com/heroku/heroku-buildpack-multi
```

#### Add Add-ons for the Heroku App

```
# Add the Heroku PostgreSQL Add-ons
# The free version works for small deployments.
heroku addons:create heroku-postgresql:hobby-dev
```

```

# Add the CloudAMQP RabbitMQ Add-on
# Odds are you'll want to upgrade this at some point in the next 15 days.
# Connect uses some AMQP resources while idle which will go past the free plan.
# But this demo assumes you want to pay $0
heroku addons:create cloudamqp:lemur

# Add the MemCachier Memcached Add-on
# Connect doesn't use much caching, but check the Add-ons for other limitations
heroku addons:create memcachier:dev

# Add the New Relic monitoring Add-on
# If you have an existing Newrelic account, skip this and add your account key as the
# `NEW_RELIC_LICENSE_KEY` environment variable and set a `NEW_RELIC_APP_NAME` variable
heroku addons:create newrelic:wayne

```

### Configure Core Connect Settings

```

# A secret key used by django
heroku config:set SECRET_KEY=random_string_here

# A secret key used just for email
heroku config:set EMAIL_SECRET_KEY=random_string

# A list of hosts you're allowing Connect to live at.
# `*` would allow ALL hostnames, but it's highly recommended you use a list such as
# 'public-connect.herokuapp.com,connect.mydomain.com'
heroku config:set ALLOWED_HOSTS=public-connect.herokuapp.com

# The hostname of the app, no protocol
heroku config:set HOSTNAME=public-connect.herokuapp.com

# The full URL of the app, with protocol.
# This will be used in outgoing emails that need an absolute URL
heroku config:set ORIGIN=https://public-connect.herokuapp.com

```

### Configure S3 Static & Upload Storage

```

# Access key, will need S3 and (if you want to use it) SES access
heroku config:set AWS_ACCESS_KEY_ID=your_aws_access_key

# Secret key to go with the above access key
heroku config:set AWS_SECRET_ACCESS_KEY=your_aws_secret_key

# Bucket you want Connect to use for static files and uploads
heroku config:set AWS_STORAGE_BUCKET_NAME=storage_bucket_to_use_here

# Folder in the S3 bucket to store uploads (change this to fit your needs)
heroku config:set DEFAULT_S3_PATH=publicconnect/uploads

# Folder in the S3 bucket to store static files
heroku config:set STATIC_S3_PATH=publicconnect/staticfiles

# Boolean to tell Connect to rely on S3
heroku config:set USE_S3=True

```

**Disable collectstatic during Heroku compilation** Connect relies on bower for static file dependencies. To keep things speedy and builds small, we need to disable the bower and collectstatic step when compiling Connect on

Heroku.

```
# Disable staticfile collection during heroku build
heroku config:set DISABLE_COLLECTSTATIC=1
```

### Deploy Connect to Heroku

```
# Push code & compile your new heroku app.
# This could take a bit, and you may have to try multiple times.
git push heroku master
```

### Setup the Connect Database

```
# Install the database
heroku run python manage.py migrate
```

### Deploy Static Assets

```
# Download, compile, and deploy static assets to S3.
# This could take a bit.
heroku run 'bower install --config.interactive=false;grunt prep;python manage.py collectstatic --noinput'
```

---

**Note:** Make sure your S3 bucket has a CORS policy, otherwise fonts will be broken. See: [Hosting Connect Fonts on Amazon S3](#)

---

### Scale up Connect

```
# Scale up the scheduler.
# You'll soon need to make the scheduler a 2x dyno, but this demo is free
heroku ps:scale web=1 scheduler=1
```

---

**Note:** There are 3 apps defined in the heroku Procfile: `web`, `scheduler` and `worker`.

You need at least 1 `web` dyno, which accepts incoming HTTP requests, and at least 1 `worker`. Your first worker must be a `scheduler` dyno (which is a [Celery](#) worker which starts [periodic tasks](#).) If your task queue is too large, you'll want to create additional `worker` dynos, which are the same as `scheduler` except they will not create new scheduled tasks.

---

**Warning:** Never create more than 1 `scheduler` dyno. If you need extra task-processing capacity, create new workers. In order to properly launch the Connect python process all worker/scheduler dynos must have at least 1GB of RAM (known as 2X dynos)

### Working with a Deployed Connect

When you visit your `https://appname.herokuapp.com` (or whatever domain you assigned Connect to) you'll have the opportunity to login via NGPVAN's ActionID. Do this, and remember the email address you used.

In order to promote your first user to a superadmin, run

```
heroku run python manage.py promote_superuser yourname@yourdomain.com
```

From here on out you'll want to follow the [Administrator Guide](#)

## Setting up Amazon's Simple Email Service

Having an outgoing email service is not required. By default Connect will discard all outgoing emails.

If you'd like to setup Amazon's Simple Email Service, change these configuration settings:

```
# Actual from address to be used. Must be whitelisted in SES
heroku config:set DEFAULT_FROM_ADDRESS=myconnectemail@mydomain.com

# The `To:` field in outgoing emails, must be SES whitelisted
heroku config:set DEFAULT_FROM_EMAIL="My Connect <myconnectemail@mydomain.com>"

# Use the "seacucumber" backend for outgoing email, which uses the task queue
heroku config:set EMAIL_BACKEND=seacucumber.backend.SESBackend
```

More detail can be found at [Outgoing Email Configuration](#)

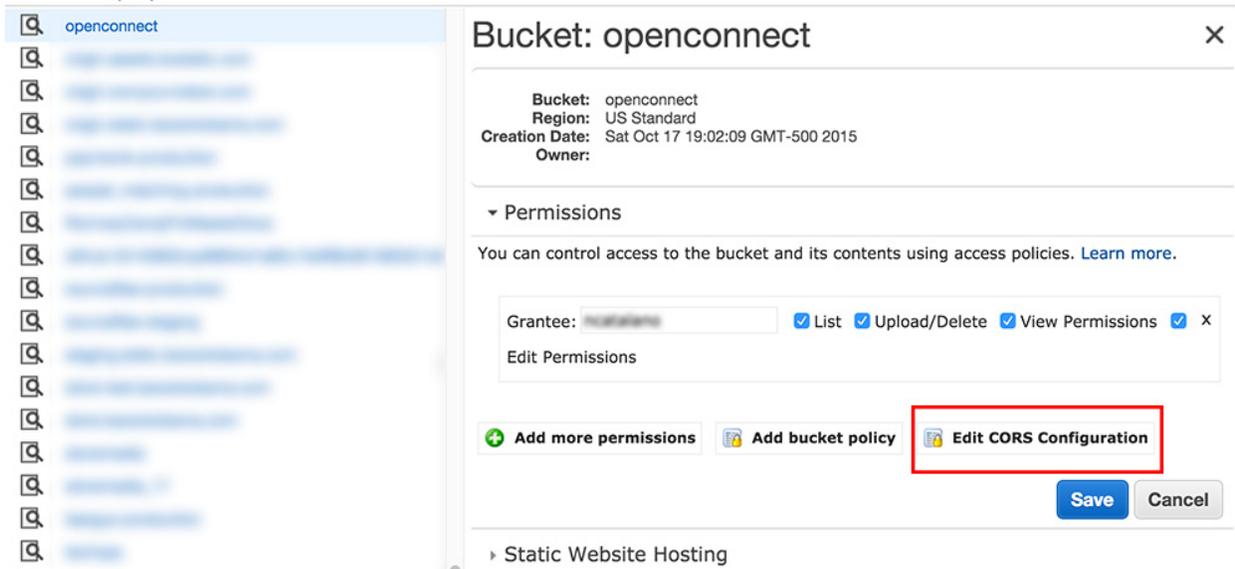
## Hosting Connect Fonts on Amazon S3

Connect relies on webfonts for some parts of the UX. Many browsers restrict the hosting of webfonts to only those with an explicit [Cross-origin resource sharing \(CORS\)](#) policy.

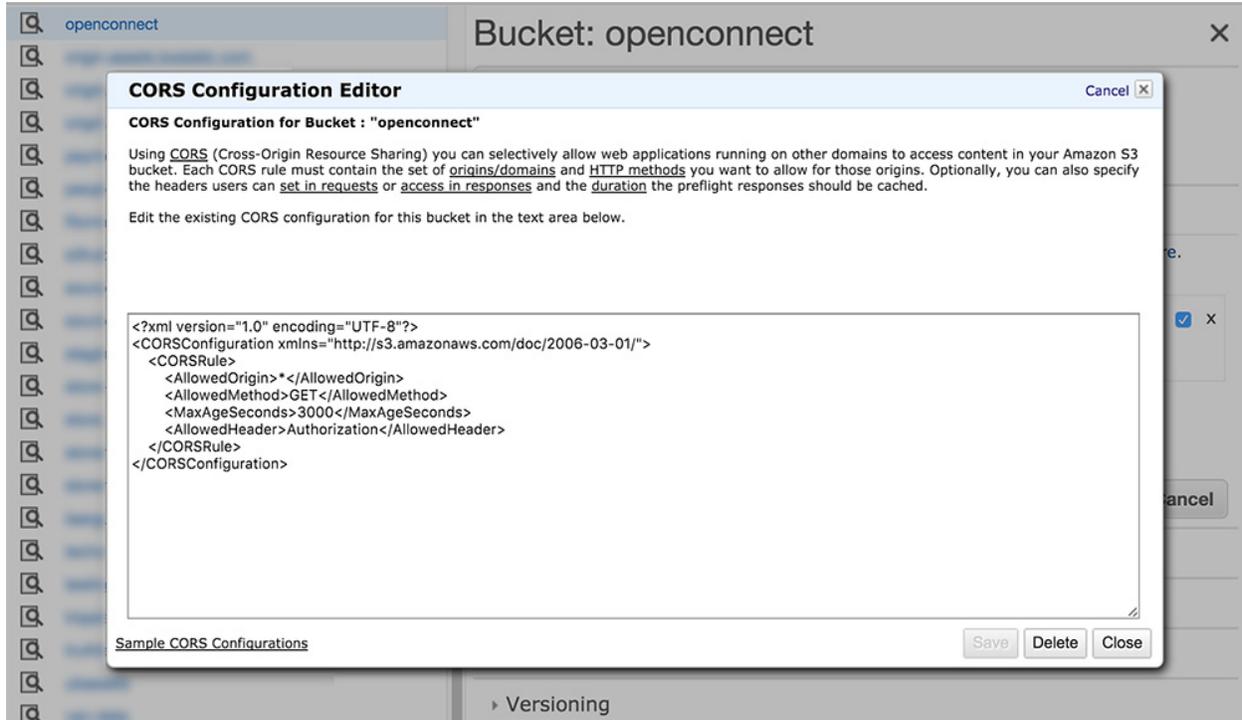
If you're relying on Amazon S3 to store static assets, you're able to assign a CORS policy per-bucket. This policy is defined using an S3-specific XML schema. For details about a custom CORS policy, view the [Amazon S3 CORS Documentation](#).

By far the easiest way to configure CORS is by [editing bucket permissions in the AWS Management Console](#).

To edit the cors configuration, go into the S3 section inside the AWS console, then click the button and press "Properties." You'll see a button labeled "Edit CORS Configuration" for you to edit.



You should see a text box pop-up in a modal which is pre-filled with a "Allow all origins to GET content" policy available by default. This should be fine for most uses.



If the textfield does not prefill, you can use this code:

```
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <CORSRule>
    <AllowedOrigin>*/AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
    <AllowedHeader>Authorization</AllowedHeader>
  </CORSRule>
</CORSConfiguration>
```

## Development and Deployment with Docker

Connect ships with support for [Docker](#).

The `Dockerfile` will install all the required dependencies for the build, as well as a script called `proclaunch`. As the `Dockerfile` does not have a `CMD` or `ENTRYPOINT` which launches all of Connect (nor does it have all the required settings pre-defined), to launch each process you'll need to run `proclaunch web`, `proclaunch scheduler`, and/or `proclaunch worker` to run the correct process.

By default there are not enough environment settings to fully launch Connect. You'll want to create a `.env` file to pass into docker which contains the settings in [Connect Settings](#).

In order to have dev/production parity, you'll want to also launch docker containers for [Memcached](#) and [RabbitMQ](#) and set the `BROKER_URL` and `CACHE_URL` to link to those. `docker-compose.yml` contains example code for running web, scheduler and worker processes.

**Note:** The first time you build the `dockerfile` (either via the standard docker build process or via `docker-compose build`) your machine will have to pull in all the dependencies for both frontend and backend code. Subsequent builds will skip these steps. If you want to re-pull dependencies, the easiest step is to change `requirements.txt` for

python packages, `package.json` for node packages, or `bower.json` for static dependencies. Docker will scan these files for changes during each build process, and if there is a change it will re-run that part of the build process.

**Warning:** While `docker-compose.yml` is a good starting-point for knowing what settings should be defined, it's written for development.

## 3.1.2 Individual Topics

### Connect Settings

- *Required*
- *Application*
- *Base*
- *Email Settings*
- *Authentication*
- *Celery/Task Queue*
- *Storage*
- *Logging*

#### Required

Without these settings set you will not be able to start Connect

#### **SECRET\_KEY No Default Provided**

From the Django Documentation:

A secret key for a particular Django installation. This is used to provide [cryptographic signing](#), and should be set to a unique, unpredictable value.

Connect will refuse to start if `SECRET_KEY` is not set.

#### **EMAIL\_SECRET\_KEY No Default Provided**

Similar to `SECRET_KEY` the Email Secret Key is a secret key that is used in cryptographic functions that involve outgoing emails.

As Connect usually requires users to be logged in to perform actions such as modify their subscription statuses we can usually verify that a specific user has permission to change these settings by comparing their login state.

In order to prevent spam complaints we do not require users to log in to unsubscribe if they are unsubscribing via a link in an email. In order to prevent a malicious third party from bulk unsubscribing users, we generate a hash based on the email address and `EMAIL_SECRET_KEY` to confirm that the user visiting the unsubscribe URL is in fact authorized to unsubscribe that user, even if they are not logged in.

This is different from `SECRET_KEY` since you may wish to keep your `SECRET_KEY` unique and secure on staging and production and require developers to have their own local `SECRET_KEY`, but you may also want to have a shared `EMAIL_SECRET_KEY` among developers and staging systems.

It's usually wise to have both be unique.

### DATABASE\_URL No Default Provided

Connect is optimized to work with PostgreSQL as the database backend on top of Heroku, with the database being provided by Heroku’s Postgres service. As a backup for local development where PostgreSQL is not available, it is possible to use SQLite, although support for all features is not guaranteed.

In order to provide database service Heroku sets an environment variable called DATABASE\_URL.

But that doesn’t mean that there isn’t flexibility built in for users who may want to use alternative database engines.

For local development using [Postgres.app](#) we usually use DATABASE\_URL=postgres://localhost/connect

Here is the URL structure from the dj-database-url package readme (note that only PostgreSQL and SQLite work out of the box, and SQLite is not recommended for production):

Engine	Django Backend	URL
PostgreSQL	django.db.backends.postgresql_psycopg2	postgres://USER:PASSWORD@HOST:PORT/NAME <sup>1</sup>
PostGIS	django.contrib.gis.db.backends.postgis	postgis://USER:PASSWORD@HOST:PORT/NAME
MySQL	django.db.backends.mysql	mysql://USER:PASSWORD@HOST:PORT/NAME
MySQL (GIS)	django.contrib.gis.db.backends.mysqlgis	mysqlgis://USER:PASSWORD@HOST:PORT/NAME
SQLite	django.db.backends.sqlite3	sqlite:///PATH <sup>2</sup>

**Warning:** Connect is tested against and run in production with PostgreSQL. SQLite has been confirmed to work in the past, although its usage is discouraged. Past issues with MySQL have included problems with out-of-the-box emoji support. Pull requests which improve MySQL support would be highly appreciated.

### CACHE\_URL No Default Provided

Connect is heavily reliant on a cache, and there is an inherent danger in choosing a default that may end up in production. As such, we require that you explicitly set a CACHE\_URL in your environment.

Engine	Django Backend	URL
Local Memory	django.core.cache.backends.locmem.LocMemCache	locmemcache:// [NAME]
Dummy	django.core.cache.backends.dummy.DummyCache	dummycache://
Database	django.core.cache.backends.db.DatabaseCache	database://USER:PASSWORD@HOST:PORT/NAME
File	django.core.cache.backends.filebased.FileBasedCache	filebasedcache://PATH/TO/FILE
memcached	django.core.cache.backends.memcached.MemcachedCache	memcachedcache://HOST:PORT
pymemcached	django.core.cache.backends.memcached.PyMemcacheCache	pymemcache://HOST:PORT
redis	django_redis.cache.RedisCache	redis:// [USER:PASSWORD@]HOST:PORT [/DB]

**Note:** Heroku provides multiple memcached add-ons which provide cache support using their own custom environment variables. By default the Heroku version of Connect will install `django-heroku-memcacheify`, which will allow you to use your choice of [MemCachier](#) or [Memcached Cloud](#) out of the box.

<sup>1</sup>With PostgreSQL, you can also use unix domain socket paths with [percent encoding](#): postgres://%2Fvar%2Flib%2Fpostgresql/dbname.  
<sup>2</sup>SQLite connects to file based databases. The same URL format is used, omitting the hostname, and using the “file” portion as the filename of the database. This has the effect of four slashes being present for an absolute file path: sqlite:///full/path/to/your/database/file.sqlite.

**Warning:** When testing Connect we recommend you use the `dummycache://` setting to avoid cross-test cache contamination. Running `CACHE_URL=dummycache:// python manage.py test` for your testing is the best way to guarantee tests are being run correctly.

**ALLOWED\_HOSTS** Default: `[]` (Empty list)

While not required while `DEBUG=True`, to run Connect in production you'll need your `ALLOWED_HOSTS` setting to be set.

From the Django Documentation:

A list of strings representing the host/domain names that this Django site can serve. This is a security measure to prevent an attacker from poisoning caches and triggering password reset emails with links to malicious hosts by submitting requests with a fake `HTTP Host` header, which is possible even under many seemingly-safe web server configurations.

Values in this list can be fully qualified names (e.g. `'www.example.com'`), in which case they will be matched against the request's `Host` header exactly (case-insensitive, not including port). A value beginning with a period can be used as a subdomain wildcard: `'example.com'` will match `example.com`, `www.example.com`, and any other subdomain of `example.com`. A value of `'*'` will match anything; in this case you are responsible to provide your own validation of the `Host` header.

Django also allows the [fully qualified domain name \(FQDN\)](#) of any entries. Some browsers include a trailing dot in the `Host` header which Django strips when performing host validation.

When `DEBUG` is `True` or when running tests, host validation is disabled; any host will be accepted. Thus it's usually only necessary to set it in production.

## Application

These are some variables that are necessary to the functionality and display of Connect, specifically in templates and emails where minimization of changes in any fork is important.

**BRAND\_TITLE** Default: `Connect` (String)

The title you're using for your version of Connect. This is used throughout the Connect codebase.

**ORGANIZATION** Default: `Owner` (String)

The name of the organization or person running this copy of Connect. This will be attached to all outgoing emails as well as included in a few copyright sections.

**HOSTNAME** Default: `localhost:8000` (String)

The hostname that this version of Connect is running on, without the protocol. If connect is running at `https://public.ofaconnect.com/` then the hostname would be `public.ofaconnect.com`

**ORIGIN** Default: `http://localhost:8000` (String)

The full URL that this version of Connect is running at, with the protocol. This is used when absolute URLs are needed (such as in notification emails.) If Connect is running at `https://public.ofaconnect.com/` this would be `https://public.ofaconnect.com`

**DEFAULT\_FROM\_ADDRESS** Default: `no-reply@connect.local` (String)

The “From” address will be used when outgoing emails are compiled by Connect.

**You must have this address whitelisted to be sent from with your Email Service Provider**

This is the raw email address, with no names attached.

**DEFAULT\_FROM\_EMAIL** Default: `Connect <no-reply@connect.local>` (String)

The friendly “From” address that will be used on outgoing emails sent from Connect. This is what will appear in your end user’s email client as the sender of notifications.

**You must have this address whitelisted to be sent from with your Email Service Provider**

**SYSTEM\_USER\_NAME** Default: `Connect` (String)

Connect has a `System User` which is the user account that Connect uses internally for notifications and other actions that need to be performed on the end-user level.

**SYSTEM\_USER\_EMAIL** Default: `connect@connect.local` (String)

It’s possible to override the email address of the system user.

**This is not important for the functioning of Connect and once set this can never be changed. So it’s wise to just leave the default.**

**Warning:** If you do decide to change this, and it is recommended you do not, realize that you’ll immediately have to change the email address of the system user in your database to reflect the new setting.

**GOOGLE\_ANALYTICS\_PROPERTY\_ID** Default: `UA-0-0` (String)

The [Google Analytics](#) property ID.

[Google Analytics support](#) has details on how to find this code.

**GOOGLE\_ANALYTICS\_DEBUG\_MODE** Default: `False` (Boolean)

A boolean specifying if Connect should set Google Analytics into [Debug Mode](#).

**This is likely only necessary to change if you’re developing Google Analytics code**

**CONNECT\_APP** Default: **No Default**

The app that contains the assets and templates for the version of Connect you’ll want to use.

If you’re branding your own version of Connect you’ll likely want to change this to `private_connect` (or whatever app name you choose)

If no custom `CONNECT_APP` is defined Connect will fall back to the open source assets and templates.

**ICON\_PREFIX** Default: `glyphicon glyphicon-` (String)

Connect offers you the ability to swap-out the standard [Glyphicon](#) library by specifying a prefix of both the class name for the iconset as well as the prefix for the icon itself.

## Base

**DEBUG** Default: `False` (boolean)

From the Django Documentation:

Never deploy a site into production with `DEBUG` turned on.

Did you catch that? NEVER deploy a site into production with `DEBUG` turned on.

One of the main features of debug mode is the display of detailed error pages. If your app raises an exception when `DEBUG` is `True`, Django will display a detailed traceback, including a lot of metadata about your environment, such as all the currently defined Django settings (from `settings.py`).

As a security measure, Django will *not* include settings that might be sensitive (or offensive), such as `SECRET_KEY`. Specifically, it will exclude any setting whose name includes any of the following:

- `'API'`
- `'KEY'`
- `'PASS'`
- `'SECRET'`
- `'SIGNATURE'`
- `'TOKEN'`

Note that these are *partial* matches. `'PASS'` will also match `PASSWORD`, just as `'TOKEN'` will also match `TOKENIZED` and so on.

Still, note that there are always going to be sections of your debug output that are inappropriate for public consumption. File paths, configuration options and the like all give attackers extra information about your server.

It is also important to remember that when running with `DEBUG` turned on, Django will remember every SQL query it executes. This is useful when you're debugging, but it'll rapidly consume memory on a production server.

Finally, if `DEBUG` is `False`, you also need to properly set the `ALLOWED_HOSTS` setting. Failing to do so will result in all requests being returned as "Bad Request (400)".

**TIME\_ZONE** Default: `US/Eastern` (String)

A string representing the time zone for this installation. It's recommended that you choose from one of the following:

- `US/Eastern`
- `US/Central`
- `US/Mountain`
- `US/Pacific`

**LANGUAGE\_CODE** Default: `en-us` (String)

A string representing the language code for this installation.

Currently Connect only supports the United States English language code (`en-us`) although if you want to adapt Connect into another language this would be a setting you'd want to change.

**SESSION\_COOKIE\_NAME** Default: `messages_sessionid` (String)

In order to avoid cookie collisions and overwrites with other applications hosted on the same domain, the session cookie name is customized on Connect. If you want to have multiple installations of Connect on the same domain where a user could be simultaneously logged into each version it's likely you'll need to change this setting.

---

**Note:** It is possible to have multiple copies of Connect on the same domain using the same `messages_sessionid` cookie name. Just realize that your browser can only be logged into one copy of Connect per domain.

---

**CSRF\_COOKIE\_NAME** Default: `messages_csrf_token` (String)

The “CSRF Cookie” is a browser cookie created by Connect that prevents third parties from using javascript to perform actions as users. This type of attack, known as a [Cross-site request forgery](#), is a large concern for Connect.

You can change the name of the cookie here. This may be important if you have multiple installations of Connect on the same domain and want to allow users to be simultaneously logged into both, although this is not recommended.

**Warning:** The Connect frontend assumes that the CSRF cookie is called `messages_csrf_token`, so changing this setting may involve finding all the references to `messages_csrf_token` in Connect's frontend code to maintain HTTP POST functionality.

**SESSION\_ENGINE** Default: `django.contrib.sessions.backends.cache_db` (String)

From the Django documentation:

Controls where Django stores session data. Included engines are:

- `'django.contrib.sessions.backends.db'`
- `'django.contrib.sessions.backends.file'`
- `'django.contrib.sessions.backends.cache'`
- `'django.contrib.sessions.backends.cache_db'`
- `'django.contrib.sessions.backends.signed_cookies'`

**SESSION\_SERIALIZER** Default: `django.contrib.sessions.serializers.PickleSerializer` (String)

Full import path of a serializer class to use for serializing session data. Included serializers are:

- `'django.contrib.sessions.serializers.PickleSerializer'`
- `'django.contrib.sessions.serializers.JSONSerializer'`

---

**Note:** While not optimal, we use `PickleSerializer` for Connect to handle some edge cases that have cropped up in the past using the `JSONSerializer`.

---

**SESSION\_EXPIRE\_AT\_BROWSER\_CLOSE** Default: `False` (String)

Whether to expire the session when the user closes their browser.

---

**Note:** There is significant advantage to setting this to `True` when using authentication backends which will “Trust” Connect and immediately authenticate users coming from Connect if the user is logged into the authentication provider. That way if the user logs out of their account with the authentication provider they’re also logged out of Connect.

---

**SESSION\_COOKIE\_SECURE** Default: `False` (String)

Whether to use a secure cookie for the session cookie. If this is set to `True`, the cookie will be marked as “secure,” which means browsers may ensure that the cookie is only sent under an HTTPS connection.

**It’s highly recommended you set this to “True“ in production**

**SECURE\_PROXY\_SSL\_HEADER** Default: `'HTTP_X_FORWARDED_PROTO', 'https'` (Tuple)

A tuple representing a HTTP header/value combination that signifies a request is secure. This controls the behavior of the request object’s `is_secure()` method.

**Warning:** This is included by default to match the header Heroku sends to signify if a request is secure or not. Heroku will not allow an end-user to spoof the `HTTP_X_FORWARDED_PROTO` header. If you’re deploying Connect on a different platform make sure it is not possible for an end user to spoof the `HTTP_X_FORWARDED_PROTO` header, otherwise set this to `None` or a different non-spoof-able header.

From the Django Documentation:

This takes some explanation. By default, `is_secure()` is able to determine whether a request is secure by looking at whether the requested URL uses “`https://`”. This is important for Django’s CSRF protection, and may be used by your own code or third-party apps.

If your Django app is behind a proxy, though, the proxy may be “swallowing” the fact that a request is HTTPS, using a non-HTTPS connection between the proxy and Django. In this case, `is_secure()` would always return `False` – even for requests that were made via HTTPS by the end user.

In this situation, you’ll want to configure your proxy to set a custom HTTP header that tells Django whether the request came in via HTTPS, and you’ll want to set `SECURE_PROXY_SSL_HEADER` so that Django knows what header to look for.

You’ll need to set a tuple with two elements – the name of the header to look for and the required value. For example:

```
SECURE_PROXY_SSL_HEADER = ('HTTP_X_FORWARDED_PROTO', 'https')
```

Here, we’re telling Django that we trust the `X-Forwarded-Proto` header that comes from our proxy, and any time its value is `'https'`, then the request is guaranteed to be secure (i.e., it originally came in via HTTPS). Obviously, you should *only* set this setting if you control your proxy or have some other guarantee that it sets/strips this header appropriately.

Note that the header needs to be in the format as used by `request.META` – all caps and likely starting with `HTTP_`. (Remember, Django automatically adds `'HTTP_'` to the start of x-header names before making the header available in `request.META`.)

**KEY\_PREFIX** Default: (Empty string)

A string that will be automatically included (prepended by default) to all cache keys used by the Django server.

---

**Note:** This would be useful to modify if you want to share one memcached cluster across multiple installations of Connect.

---

### Email Settings

The ability for Connect to send outgoing email is vital. It's highly recommended you read [Outgoing Email Configuration](#) before attempting to configure outgoing email in production.

**USE\_SES** Default: `False` (Boolean)

A boolean specifying if Connect should use Amazon's [Simple Email Service](#).

**EMAIL\_BACKEND** Default: `django.core.mail.backends.dummy.EmailBackend` (String)

The outgoing email backend that Connect should use.

If using the SMTP backend, this will need to be set to `django.core.mail.backends.smtp.EmailBackend`.

If you're using SES you'll likely want to use [Sea Cucumber](#), a Django email backend library that will use [Celery](#) to queue and rate-limit outgoing Simple Email Service requests, and thus set this to `seacucumber.backend.SESBackend`.

**EMAIL\_HOST** Default: `localhost` (String)

#### For the SMTP backend only

The host to use for sending email via SMTP.

**EMAIL\_HOST\_PASSWORD** Default: (Empty string)

#### For the SMTP backend only

Password to use for the SMTP server defined in `EMAIL_HOST`. This setting is used in conjunction with `EMAIL_HOST_USER` when authenticating to the SMTP server. If either of these settings is empty, Django won't attempt authentication.

**EMAIL\_HOST\_USER** Default: (Empty string)

#### For the SMTP backend only

Username to use for the SMTP server defined in `EMAIL_HOST`. If empty, Django won't attempt authentication.

**EMAIL\_PORT** Default: `25` (Integer)

#### For the SMTP backend only

Port to use for the SMTP server defined in `EMAIL_HOST`.

**EMAIL\_USE\_TLS** Default: `False` (Boolean)

#### For the SMTP backend only

Whether to use a TLS (secure) connection when talking to the SMTP server. This is used for explicit TLS connections, generally on port 587. If you are experiencing hanging connections, see the implicit TLS setting `EMAIL_USE_SSL`.

**EMAIL\_USE\_SSL** Default: `False` (Boolean)

**For the SMTP backend only**

Whether to use an implicit TLS (secure) connection when talking to the SMTP server. In most email documentation this type of TLS connection is referred to as SSL. It is generally used on port 465. If you are experiencing problems, see the explicit TLS setting `EMAIL_USE_TLS`.

Note that `EMAIL_USE_TLS/EMAIL_USE_SSL` are mutually exclusive, so only set one of those settings to `True`.

**EMAIL\_SSL\_CERTFILE** Default: `None` (String)

**For the SMTP backend only**

If `EMAIL_USE_SSL` or `EMAIL_USE_TLS` is `True`, you can optionally specify the path to a PEM-formatted certificate chain file to use for the SSL connection.

**EMAIL\_SSL\_KEYFILE** Default: `None` (String)

**For the SMTP backend only**

If `EMAIL_USE_SSL` or `EMAIL_USE_TLS` is `True`, you can optionally specify the path to a PEM-formatted private key file to use for the SSL connection.

Note that setting `EMAIL_SSL_CERTFILE` and `EMAIL_SSL_KEYFILE` doesn't result in any certificate checking. They're passed to the underlying SSL connection. Please refer to the documentation of Python's `python:ssl.wrap_socket()` function for details on how the certificate chain file and private key file are handled.

**EMAIL\_TIMEOUT** Default: `None` (Integer)

**For the SMTP backend only**

Specifies a timeout in seconds for blocking operations like the connection attempt.

**CUCUMBER\_RATE\_LIMIT** For the Sea Cucumber backend only

Default: `1` (Integer)

The number of emails per second that will be sent from Connect via Sea Cucumber.

If you are a new SES user, your default quota will be 1,000 emails per 24 hour period at a maximum rate of one email per second. You can use the command `python manage.py ses_usage` to get your quota.

**BOUNCY\_AUTO\_SUBSCRIBE** Default: `False` (Boolean)

Used by the [Django Bouncy](#) library.

All Amazon [Simple Notification Service](#) (SNS) endpoints must verify with Amazon that they're willing to accept incoming messages. Setting `BOUNCY_AUTO_SUBSCRIBE` to `True` will tell Connect to verify with Amazon any incoming SNS subscription requests.

In order to avoid malicious third parties with Amazon Web Services accounts from sending unsubscribe requests to your version of Connect, this is turned off by default.

**BOUNCY\_TOPIC\_ARN** Default: None (List)

Used by the [Django Bouncy](#) library.

All Simple Notification Service queues are assigned a unique [Amazon Resource Name \(ARN\)](#). Connect allows you to specify a list of valid ARNs that should be allowed to unsubscribe users.

Considering `BOUNCY_AUTO_SUBSCRIBE` is set to `False` Connect should never subscribe to a malicious third party's notification queue in the first place, but for added assurance it may make sense to add your ARN to this setting.

### Authentication

Connect relies heavily on [Python Social Auth](#) for authentication.

**DEFAULT\_AUTH\_BACKEND** Default: `social.backends.ngpvan.ActionIDOpenID` (String)

Also available: `connect_extras.auth_backends.bsdttools.BSDToolsOAuth2`

You can find out more information about different authentication backends available at [Authentication Backends](#)

**POST\_LOGOUT\_PAGE** Default: `/` (String)

If `DEFAULT_AUTH_BACKEND` is `social.backends.ngpvan.ActionIDOpenID` this defaults to `https://accounts.ngpvan.com/Account/Logout`

If `DEFAULT_AUTH_BACKEND` is `connect_extras.auth_backends.bsdttools.BSDToolsOAuth2` this defaults to `https://{BSDTOOLS_INSTANCE}/page/user/logout`

**SOCIAL\_AUTH\_NEW\_USER\_REDIRECT\_URL** Default: `/explore/` (String)

**LOGIN\_REDIRECT\_URL** Default: `/messages/` (String)

**LOGIN\_ERROR\_URL** Default: `/` (String)

**SOCIAL\_AUTH\_PROTECTED\_FIELDS** Default: `username,` (List)

**USE\_SOCIAL\_AUTH\_AS\_ADMIN\_LOGIN** Default: `True` (Boolean)

**OPTIONAL: BSDTOOLS\_INSTANCE** Default: (Empty string)

**OPTIONAL: BSDTOOLS\_KEY** Default: (Empty string)

**OPTIONAL: BSDTOOLS\_SECRET** Default: (Empty string)

## Celery/Task Queue

Connect relies on [Celery](#) as a distributed task queue and scheduler.

Connect specifically uses tasks when actions a) do not need to be completed immediately, b) are especially intensive or lengthy, or c) to be run automatically on a schedule.

While there is no expectation that tasks will be performed instantly, users will notice if tasks are substantially backed up and as such you should ensure that you have enough workers assigned to promptly handle new tasks.

**BROKER\_URL** Default: `django://` (String)

Also allowed: `CLOUDAMQP_URL`

Connect is mostly broker-agnostic, but the default implementation uses [RabbitMQ](#).

The expected format of the `BROKER_URL` for different backends can be found in the [Celery Broker Documentation](#).

In order support the [CloudAMQP Heroku Addon](#) if `BROKER_URL` is not present and `CLOUDAMQP_URL` is, Connect will use `CLOUDAMQP_URL` as the broker URL.

**BROKER\_POOL\_LIMIT** Default: `1` (Integer)

**BROKER\_HEARTBEAT** Default: `30` (Integer)

**BROKER\_CONNECTION\_TIMEOUT** Default: `30` (Integer)

**CELERY\_EVENT\_QUEUE\_EXPIRES** Default: `60` (Integer)

**CELERY\_ALWAYS\_EAGER** Default: `True` (Boolean)

**CELERY\_TIMEZONE** Default: `UTC` (String)

**CELERY\_SEND\_EVENTS** Default: `False` (Boolean)

**CELERY\_RESULT\_BACKEND** Default: `None` (String)

**ALSO: CLOUDAMQP\_URL** Default: (Empty string)

## Storage

By default Connect relies on [Amazon S3](#) for storage functionality.

**AWS\_ACCESS\_KEY\_ID** Default: (Empty string)

An Amazon Web Services access key id. This is also used by the Sea Cucumber library for outgoing email via Simple Email Service should you enable that functionality.

**AWS\_SECRET\_ACCESS\_KEY** Default: (Empty string)

The associated secret key associated with the `AWS_ACCESS_KEY_ID`

**USE\_S3** Default: `False` (Boolean)

A boolean set to `True` if Connect is to store content on Amazon S3.

**AWS\_STORAGE\_BUCKET\_NAME** Default: (Empty string)

The name of a storage bucket that the `AWS_ACCESS_KEY_ID` has full access to and can upload both static assets and media for Connect.

**DEFAULT\_S3\_PATH** Default: `connect/uploads` (String)

The default path in the S3 bucket for uploads to be uploaded to.

**Warning:** If you want to use the same S3 bucket for multiple Connect installations, such as a staging and production installation, you should make `DEFAULT_S3_PATH` and `STATIC_S3_PATH` unique.

**STATIC\_S3\_PATH** Default: `connect/static` (String)

The default path in the S3 bucket for static files to be uploaded to.

### Logging

**LOG\_LEVEL** Default: `WARNING` (String)

### Outgoing Email Configuration

Connect is heavily reliant on outgoing email notifications. Incoming email replies are not supported yet. Making sure those emails are sent by Connect (and reliably received by the end-user) means it's necessary to link Connect to a 3rd party transactional email service provider.

#### Email Service Provider

The recommended outgoing email service provider for Connect is Amazon's [Simple Email Service](#) (or SES.)

Connect comes pre-installed with the [Sea Cucumber](#) library, which will put all outgoing messages into Connect's task queue, allowing your Heroku workers to stagger their outgoing email to match the send speed limits spelled out on Amazon S3.

To use the Sea Cucumber library, make sure your `AWS_ACCESS_KEY_ID` and `AWS_SECRET_ACCESS_KEY` settings are correct and have the correct SES permissions and then set the `EMAIL_BACKEND` environment variable to `seacucumber.backend.SESBackend`

**Warning:** Connect does not support incoming email. Users who reply to emails from Connect such as new message digests and notifications will receive the (quite ugly) "Your email could not be delivered" reply by default. It's recommended you use an outgoing address like `no-reply@yourdomain.com` and have that address reply with a friendly "Your email was not received, please visit Connect to comment."

---

## Handling Bounces

Connect has a built-in ‘Unsubscribe’ feature. Each outgoing message checks the `open_connect.mailer.models.Unsubscribe` table for a record that matches the outgoing email address. If a record is found, the outgoing message is discarded.

Installed by default is OFA’s in-house-built [Django Bouncy](#) application, which will allow you to send bounces and spam complaints from SES directly into Connect via Amazon’s [Simple Notification Service](#) (or SNS.) Hard bounces and complaints are configured to create `Unsubscribe` objects in Connect, which will result in the app being denied.

More details about configuring Django Bouncy can be found in the [Django Bouncy Readme](#). In order to prevent nefarious third parties from bulk un-subscribing connect members, by default `BOUNCY_AUTO_SUBSCRIBE` is set to `False` and `BOUNCY_TOPIC_ARN` is empty. Both of these can be overridden by setting their relevant environment variables.

The path Django Bouncy is installed at on Connect is `https://connect.yourdomain.local/mail/bouncy/`.

---

**Note:** It’s important that the Django Bouncy path is included in the `LOGIN_EXEMPT_URLS` list in the `connect.settings.authentication_settings` settings file. Otherwise Connect will attempt to forward Amazon SNS to your login provider. By default `mail/*` is in that list.

---

## Authentication Backends

Connect does not have a built-in authentication system and instead relies on the [Python Social Auth](#) library and third party authentication providers.

### NGP VAN ActionID

By default Connect uses the [NGP VAN ActionID](#) OAuth1 authentication backend. ActionID is the same backend that is used as part of NGP VAN’s VAN product, allowing users to use the same username and password to login to both VAN and Connect.

Using ActionID requires no work on your end, however there will be an opt-in step after users first attempt to login to your version of Connect using ActionID.

NGP VAN is able to “white list” certain installations of Connect based on hostname. This is already done for `http://localhost`, but by contacting [NGP VAN Developer Support](#) you may be eligible to have that authorization step removed.

### Blue State Digital Tools

Connect ships with built-in support for [Blue State Digital](#)’s [BSDTools](#) tool-set, via BSDTools’ OAuth2 provider service. This allows users to login to Connect using the same login that they use to login to the BSDTools event platform.

---

**Note:** It’s recommended you contact Blue State Digital support for more information regarding implementing an OAuth2 consumer for Connect.

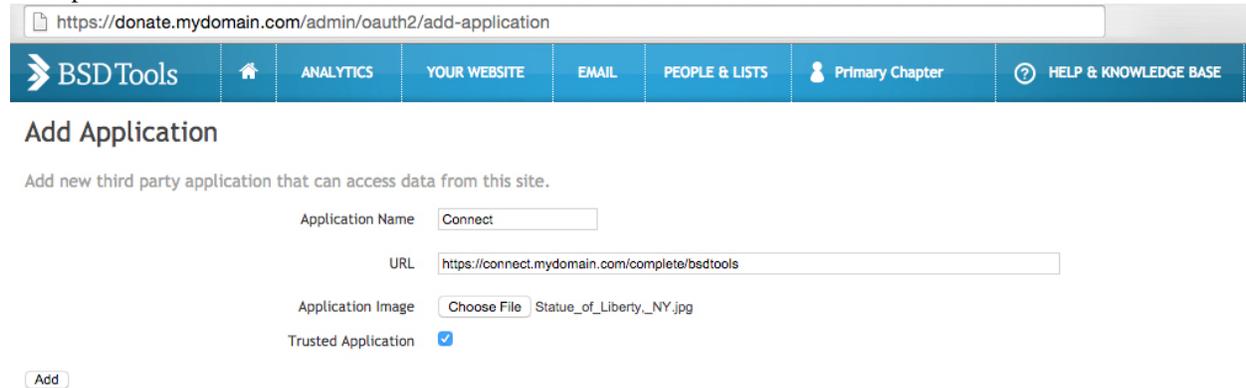
---

**Adding OAuth2 Applications to BSDTools** To create a new endpoint, go to the “Add Application” page (located at <https://bsd.clientdomain.com/admin/oauth2/add-application>) and enter your “Application Name” as “Connect”, “URL” as the full secure hostname of your BSD installation

When creating a new OAuth2 application you’ll find 4 fields necessary:

- **Application Name** The internal name that this OAuth2 application will be referred to as within BSDTools
- **URL** The “Redirect Url” that BSDTools will send users to after they’ve successfully authenticated. This must be in the format (assuming your copy of Connect is located at <https://connect.yourdomain.com>) <https://connect.yourdomain.com/complete/bsdtools>
- **Application Image** A thumbnail used by BSDTools and displayed to the user if the application is not marked as “Trusted.” Not usually publicly viewable.
- **Trusted Application** If an application is marked as “Trusted” BSDTools will immediately log a user into Connect after they log into the tools. In addition, if the user is already authenticated on BSDTools (perhaps after using BSDTools’ event functionality) they will be immediately and transparently logged into Connect if this is checked.

Example:



<https://donate.mydomain.com/admin/oauth2/add-application>

**BSDTools** | [ANALYTICS](#) | [YOUR WEBSITE](#) | [EMAIL](#) | [PEOPLE & LISTS](#) | [Primary Chapter](#) | [HELP & KNOWLEDGE BASE](#)

### Add Application

Add new third party application that can access data from this site.

Application Name

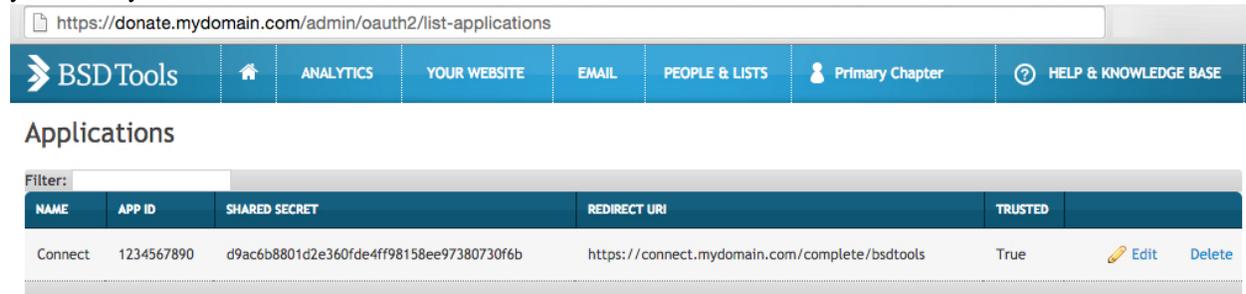
URL

Application Image  Statue\_of\_Liberty\_NY.jpg

Trusted Application

**Note:** If you’re testing Connect locally with a port (such as <http://localhost:8000>) you’ll need to set the URL to <http://localhost:8000/complete/bsdtools>

**Listing Available OAuth2 Applications in BSDTools** Once you’ve created your application you’ll need the APP ID and SHARED SECRET. You can find this by going to the “List Applications” page (located at <https://bsd.clientdomain.com/admin/oauth2/list-applications>) and finding the application you recently created.



<https://donate.mydomain.com/admin/oauth2/list-applications>

**BSDTools** | [ANALYTICS](#) | [YOUR WEBSITE](#) | [EMAIL](#) | [PEOPLE & LISTS](#) | [Primary Chapter](#) | [HELP & KNOWLEDGE BASE](#)

### Applications

Filter:

NAME	APP ID	SHARED SECRET	REDIRECT URI	TRUSTED	
Connect	1234567890	d9ac6b8801d2e360fde4ff98158ee97380730f6b	<a href="https://connect.mydomain.com/complete/bsdtools">https://connect.mydomain.com/complete/bsdtools</a>	True	<a href="#">Edit</a> <a href="#">Delete</a>

**Configuring Connect to use BSDTools** You'll need to then update Connect with the following settings:

- `DEFAULT_AUTH_BACKEND` should be set to `connect_extras.auth_backends.bsdttools.BSDToolsOAuth2`
- `BSDTOOLS_INSTANCE` should be set to the secure hostname of your BSDTools installation (i.e. `donate.mydomain.com`)
- `BSDTOOLS_KEY` should be set to your `APP_ID`
- `BSDTOOLS_SECRET` should be set to your `SHARED_SECRET`

If you're relying on BSDTools authentication for local development, you'll need to make sure that individual developers are using the same hostnames for testing, as the URL you enter in the Add-Application step will change.

**Warning: Never merge or de-dupe constituents who have linked Connect accounts.** Connect user accounts are linked to BSDTools Constituent ID numbers, so some actions Administrators can take may trigger BSDTools' de-duping systems which can cause Connect users to lose access to their accounts and make it impossible for them to create new accounts.

Be extremely careful when importing new constituents or modifying constituent primary email accounts in your BSDTools administration panel so that you do not attempt to re-import Connect users. And **never use the `merge_constituents_by_email` or `merge_constituents_by_id` API calls on Connect accounts**

### Management Command: `promote_superuser`

You can assign an existing user to be "Staff" and have all permissions (be a superuser) by running the `promote_superuser` management command and provide the email address of the user you wish to promote to a staff member with all permissions.

```
python manage.py promote_superuser email@email.com
```

If using Heroku, you'd go to your Connect directory and run

```
heroku run python manage.py promote_superuser email@email.com -a connect-app-name
```

## 3.2 Developing Connect

### 3.2.1 Local Development

#### Setting Up a Developer Environment

Connect is an application with extensive compiled less/hogan/javascript front-end and interpreted Python/Django back-end components. While it's not necessary for a back-end developer to know how to build front-end tests, nor does a front-end developer need to know how to write Python code, it is vital that anyone working on Connect have the packages for both available locally.

Some parts of this documentation are based off the [DjangoCMS Documentation](#)

---

**Note:** You will need `sudo` access to install some of these packages.

---



---

**Note:** These instructions apply for development using OS X, although there shouldn't be much change needed to develop on Linux.

---

### System Package Requirements

As a [12 Factor App](#), developers of Connect should strive for [Dev/prod parity](#), where the environment run locally is as close as possible to production.

Before starting, make sure you have [Homebrew](#) installed so you can install the required libraries. The following instructions imply that you have `brew` installed.

- [Node](#) - Command-line javascript system. `brew install node`
- [PostgreSQL](#) - Database. `brew install postgresql` or via [Postgres.app](#) with the [command line tools](#) configured. [Postgres.app](#) is more user-friendly, but both work.
- [Grunt](#) - Front-end task runner. `sudo npm install -g grunt-cli`
- [Bower](#) - Front-end dependency management. `sudo npm install -g bower`
- [libjpeg](#) - JPEG image support. `brew install libjpeg`
- [pip](#) - Python dependency management. `sudo easy_install pip`
- [virtualenv](#) or [virtualenvwrapper](#) - Python package isolation. It's recommended you install the more user-friendly [virtualenvwrapper](#), which also installs [virtualenv](#), but these instructions imply you're using the vanilla [virtualenv](#). `sudo pip install virtualenv`
- [Git](#) - Code management. This may be installed already on your system, but for the latest version of Git: `brew install git`

Optional (but recommended) packages

- [libmemcached](#) Library for compiling memcached client. `brew install libmemcached`
- [Memcached](#) - Caching framework. Note: Django offers alternative caches `brew install memcached`
- [Gifsicle](#) - Animated GIF support. Gifsicle is required for GIFs to maintain animation after resize. It also allows some tests that are otherwise skipped to run. `brew install gifsicle`
- [RabbitMQ](#) - Task Queue. Note: For convenience it's recommended you run Connect in development with `CELERY_ALWAYS_EAGER` enabled (it is enabled by default when in `DEBUG` mode), disabling local task queues. It's also possible to use alternative task queues locally. `brew install rabbitmq`

### Forking Connect

Connect requires users to fork their own version of the [Connect Github project](#) then use the official project as a remote that changes can be merged in from.

How to fork Connect and move the upstream changes into that fork is outside the scope of this documentation, but for simplicity's sake we can clone Connect from the official repository locally.

To clone the Connect project locally, set the official project as a remote called "open" and enter that folder, run the following:

```
git clone -o open https://github.com/ofa/connect.git
cd connect
```

---

**Note:** Be careful merging in upstream changes from the official Connect project into your local project, and always review code changes and commit messages manually before merging. While we strive for stability, Connect is under active development and some changes to the core project can break your local modifications.

---

## Back-end Configuration & Setup

**Installing in a virtualenv using pip** Installing inside a [virtualenv](#) is the preferred way to install any Django installation. These instructions imply you're using the vanilla version of virtualenv and not [virtualenvwrapper](#), which has more user-friendly shortcuts when dealing with virtualenv, but is slightly more difficult to setup.

To create a new virtualenv

```
virtualenv connectenv
```

**Note:** If you are *not* using a system-wide install of Python (such as with Homebrew), omit the usage of `sudo` when installing via `pip`.

Switch to the virtualenv at the command line by typing:

```
source connectenv/bin/activate
```

**Note:** Any time you want to load Connect you'll need to enter `source connectenv/bin/activate` to enter the virtual environment containing your Connect dependencies.

Connect relies on `pip` for python dependency management.

The python dependencies necessary for development of Connect are located in the `dev-requirements.txt` file <sup>1</sup>. To install all the packages necessary to run Connect, run:

```
pip install -r dev-requirements.txt
```

**Warning:** There are a few packages that are compiled during this step that require packages defined in *System Package Requirements*, specifically `libjpeg` for JPEG image handling and `PostgreSQL` (with the `PostgreSQL` command line tools) for database handling. Make sure you have installed both before attempting to install Connect.

**Setting up a .env file** Basic configuration of Connect is based around the 12 Factor [Environment-Based Configuration](#) philosophy. Instead of having to directly edit your environment, Connect's backend uses [Django-environ](#) and frontend uses [dotenv](#) to allow users to store the configuration in an `.env` file locally that is not tracked by version control. The `.env` file is a key/value file containing variables that will be loaded into the environment at startup.

The first step in setting up your developer environment is to clone the `.env-dev-example` file (which is tracked in version control) to be your local `.env` file (which will not be tracked by version control)

```
cp .env-dev-example .env
```

You can then edit the `.env` file to reflect the settings you need locally. Available settings are available in the [Connect Settings](#) documentation.

**Setting up a database** This assumes that you have [Postgres.app](#) installed and running and have correctly installed the [command line tools](#).

The `.env` file that ships with Connect assumes that you have a database called `connect` in your localhost database. To both create this database and have Connect insert all the preliminary code, run:

<sup>1</sup> There are multiple `requirements.txt` files in Connect, including an actual `requirements.txt`, which has packages aimed specifically at Heroku and may not compile on OS X. Each of these files include `common-requirements.txt`, which contains the core cross-platform packages necessary to run Connect on any platform. For development, use `dev-requirements.txt`

```
createdb connect
python manage.py migrate
```

### Front-end Configuration & Setup

**Loading front-end dependencies** Some files necessary for managing Connect are not contained in the repository itself, and instead must be brought in via the [Node Package Manager](#) and [Bower](#).

To load all front-end dependencies run:

```
npm install
bower install
```

**Compiling front-end files** Connect uses [Grunt](#) to compile static assets. A simple default task is already created to compile all the necessary static assets.

```
grunt
```

---

**Note:** Grunt is used elsewhere in Connect for front-end related tests. As long as you do not have a `CONNECT_APP` loaded in your `.env` file, you can run tasks like `grunt jasmine` and have the app-wide tests run. If you do have a `CONNECT_APP` defined in your `.env` you can run those same tasks by settings a `--target "open_connect/connect_core"` flag to your grunt tasks. For jasmine tests that would be `grunt --target "open_connect/connect_core" jasmine`

---

### Setting up your version of Connect

**Running Connect locally** Once you have your static files built, you'll be able to launch your version of Connect using Django's built-in development server.

While inside your local virtualenv (i.e. after running `source connectenv/bin/activate`) run

```
python manage.py runserver
```

You should now be able to visit `http://127.0.0.1:8000/` and see a fully functioning version of Connect using the open source theming.

**Promote a user to be a superuser** Connect uses *NGP VAN\_'s* [ActionID](#) single-sign-on system for authentication via [Python Social Auth](#).

When you first go to your development server you'll be given 2 buttons, one to Login and one to Create a New Account. Click on either and follow the login or registration flow presented by ActionID. **Remember the email address you use.** After you're done with that you'll be redirected back to Connect with a new account.

In order to upgrade your account to be a super-user, you'll need to use the `promote_superuser` Django management command and include the email address you used to sign-up for ActionID with.

```
python manage.py promote_superuser youremail@here.com
```

Your account should now be properly promoted to a superuser. From here on out you can manage your local version of Connect by following the [Administrator Guide](#).

## Developing Connect with Docker Compose

The quickest way to get Connect running locally is by using [Docker Compose](#). A `docker-compose.yml` file suitable for **development** is included in the repository.

Each time you want to re-build connect and test things out, run `docker-compose build` then `docker-compose up`.

**Warning:** The default `docker-compose.yml` file has the `DEBUG` variable set to `True` and both the Secret and Email Key baked-in. Keeping these defaults esp. As such, **do not deploy Connect to production with the settings defined in the default `docker-compose.yml` file**

### 3.2.2 Customization

#### Basic Theming

##### Initial Theming

The easiest way to give Connect a look-and-feel that matches your organization's brand while maintaining the ability to receive updates is to skin it as follows:

1. Install the needed dependencies by running, from the root directory:

```
npm i
bower i
```

Dev dependencies for Connect are managed by NPM, and front-end dependencies are managed by bower. These dependencies are required before compilation.

A top level folder called `/private_connect/` is provided as a template. You can rename this folder to anything you like, but we'll reference it with this name for the remainder of this documentation, so substitute your own folder name as needed.

This is where you will create your additional assets.

2. In your `.env` file, include the following: `CONNECT_APP=private_connect`
3. In the folder `/private_connect/assets/less`, you'll see a file called `private_config.less` – open it in your favorite editor and make changes. If you've worked with Bootstrap, this should seem somewhat familiar. It's essentially a variables file– change colors, point it to the correct images, and update as you like. You can also add overrides and additional style (either within this file, or imported from another file) as needed.
4. Compile the customized assets: from the root directory, run `grunt`.

What's happening in this step:

Grunt will use the `CONNECT_APP` assignment you set in the `.env` file to know where to look for its configuration. In this case, `/private_connect/grunt_config.json`.

There are already some default tasks setup to compile your new customized Connect assets. The `LESS` task (`/private_connect/grunt/less.js`) uses `/private_connect/assets/less/private_connect.less` as its main file and compiles out to a CSS file in `/private_connect/static/connect/css`.

Django also uses the `CONNECT_APP` assignment you set in the `.env` file to know where to look for static files, so it loads them into the app.

When you run the server and look at Connect, you should see it using your custom LESS changes instead of the default styles.

### Multiple Theme Instructions

Through the use of environment variables and a special grunt command, it's possible to use the same central repository and have multiple themes of Connect.

You can pass a `target` option to grunt at the command line so you can switch easily between tasks for different versions of Connect, with different settings, compiling in different places.

Example:

```
grunt --target="private-connect"
```

You can also write your own grunt configurations for any of the other tasks, i.e. uglification, template precompilation with hogan, etc, or add new custom tasks. Anything you can do with a normal grunt file, you can do here. See [this article](#) for more about Grunt configurations that are loaded from separate files.

### 3.2.3 Project Layout

#### Backend Project Layout

#### Frontend Project Layout

Configuration of the runtime version of Connect is done primarily via environment variables.

---

## Contributing to Connect

---

Connect is an open-source project released under the [MIT license](#) and pull requests and issues can be filed on the [Connect Github Repository](#).

### 4.1 Contribute Front-End Code

1. Install the needed dependencies by running, from the root directory:

```
npm i bower i
```

Dev dependencies for Connect are managed by NPM, and front-end dependencies are managed by bower. These dependencies are required before compilation.

2. Review Connect's client-side code: The files are found in the top level folder *assets*. Javascript, JS templates, and LESS files are located here.
3. Compile the assets: From the root directory, run the command *grunt* – this will pre-compile the Hogan templates, compile the LESS, minify the javascript, autoprefix the compiled CSS files, and copy the webfonts to the correct directory.

As a default, the Gruntfile is pointed at the default tasks configurations, location in */open\_connect/connect\_core/*. Each task has its own file in a folder called *grunt*, and global variables for use in the tasks are defined in *grunt\_config.json*.

As you make edits default client side assets, you can compile to see the changes.

**NOTE: This process is only for making open source contributions. Changing the default assets is not the recommended way of customizing/skinning Connect, as your changes would be overwritten as you upgrade versions. To customize Connect's styles, see instructions below.**

A *watch* task is provided in case you would like files to compile while you work.

### 4.2 Contribution Logistics & CLA

Before code can be accepted by Organizing for Action for inclusion in our projects, contributors must sign OFA's Contributor License Agreement.

Individuals should sign the [Individual Contributor License Agreement](#) and if your work was done as part of your employment you will need to submit the [Entity Contributor License Agreement](#).

There is also an [OFA Contributor License FAQ](#) which provides further details about contributing code. Further questions can be sent to [cla@barackobama.com](mailto:cla@barackobama.com).



---

## Connect License

---

The MIT License (MIT)

Copyright (c) 2015 Organizing for Action

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.