

---

# **django-intercom Documentation**

***Release 0.0.13***

**Ken Cochrane**

**Jun 16, 2023**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Usage</b>	<b>5</b>
<b>3</b>	<b>Enable Secure Mode</b>	<b>7</b>
<b>4</b>	<b>Intercom Inbox</b>	<b>9</b>
<b>5</b>	<b>Custom Data</b>	<b>11</b>
<b>6</b>	<b>Settings</b>	<b>13</b>
6.1	INTERCOM_APPID . . . . .	13
6.2	INTERCOM_SECURE_KEY . . . . .	13
6.3	INTERCOM_INCLUDE_USERID . . . . .	13
6.4	INTERCOM_ENABLE_INBOX . . . . .	14
6.5	INTERCOM_ENABLE_INBOX_COUNTER . . . . .	14
6.6	INTERCOM_INBOX_CSS_SELECTOR . . . . .	14
6.7	INTERCOM_CUSTOM_DATA_CLASSES . . . . .	14
<b>7</b>	<b>People who have worked on this project</b>	<b>15</b>
<b>8</b>	<b>TODO</b>	<b>17</b>
<b>9</b>	<b>Indices and tables</b>	<b>19</b>



Contents:



# CHAPTER 1

---

## Installation

---

1. Install django-intercom using pip:

```
pip install django-intercom
```

2. Add intercom to your INSTALLED\_APPS in your django settings file:

```
INSTALLED_APPS = (  
    # all  
    # other  
    # apps  
    'django_intercom',  
)
```

3. Add INTERCOM\_APPID setting to your django settings file with your intercom application Id.  
in settings.py:

```
INTERCOM_APPID = "your appID"
```





## CHAPTER 2

---

### Usage

---

Add the template tag code into your base template before the body tag.

At the top of the page put this:

```
{% load intercom %}
```

At the bottom of the page before the `</body>` tag put this:

```
{% intercom_tag %}
```



## CHAPTER 3

---

### Enable Secure Mode

---

This is optional, if it isn't set, then you will not use secure mode.

If you want to turn on secure mode, you can add `INTERCOM_SECURE_KEY` to your `settings.py` with the private key you can get from your intercom->app->security page.

in `settings.py`:

```
INTERCOM_SECURE_KEY = "your security_code"
```

You will need to look in the code samples to find the security key.

You will also need to make sure you check the “Enable secure mode” check box on the security page before this will work correctly.



## CHAPTER 4

---

### Intercom Inbox

---

Intercom has the ability to add an inbox link to your app so that people can contact you, and for you to let them know when they have a message waiting. If you would like to use these features you need to do the following.

1. Add the intercom css id to any inline element containing text, for example:

```
<a id="Intercom" href="#">Support</a>
```

2. Add the appropriate CSS to your style sheet.

#### No Icon:

```
#Intercom {  
  display: inline-block;  
  text-decoration: underline;  
  padding: 0;  
}
```

#### White Envelope (white text on black background):

```
#Intercom {  
  display: inline-block;  
  text-decoration: underline;  
  padding: 0 0 0 24px;  
  background: transparent url(https://www.intercom.io/images/white_env.png) no-repeat,  
  ↪left center;  
}
```

#### Black Envelope (black text on white/grey background):

```
#Intercom {  
  display: inline-block;  
  text-decoration: underline;  
  padding: 0 0 0 24px;
```

(continues on next page)

(continued from previous page)

```
background: transparent url(https://www.intercom.io/images/black_env.png) no-repeat,
↪left center;
}
```

If you want to show the unread message count then also add the following:

```
#Intercom em {
display: inline-block;
font-style: normal;
text-decoration: underline;
}
```

3. Configure your settings. Add the following to your django settings if you would like to change the defaults.

#### **INTERCOM\_ENABLE\_INBOX**

Default: True

In settings.py:

```
INTERCOM_ENABLE_INBOX = True
```

#### **INTERCOM\_ENABLE\_INBOX\_COUNTER**

Default: True

In settings.py:

```
INTERCOM_ENABLE_INBOX_COUNTER = True
```

#### **INTERCOM\_INBOX\_CSS\_SELECTOR**

Default: '#Intercom'

In settings.py:

```
INTERCOM_INBOX_CSS_SELECTOR = '#Intercom'
```

## CHAPTER 5

---

### Custom Data

---

Intercom.io allows you to send them your own custom data, django-intercom makes this easy. All you need to do is create a Class with a custom\_data method that accepts a Django User model as an argument and returns a dictionary. Here is an example:

```
from thepostman.models import message

class IntercomCustomData:
    """ Custom data class located anywhere in your project
        This one is located in thepostman/utils/custom_data.py """

    def custom_data(self, user):
        """ Required method, same name and only accepts one attribute (django User_
        ↪model) """

        num_messages = message.objects.filter(user=user).count()
        num_unread = messages.objects.filter(user=user, read=False).count()

        return {
            'num_messages' : num_messages,
            'num_unread' : num_unread,
        }
```

Once you have your classes built, you will need to register them with django-intercom so that it knows where to find them. You do this by adding the class to the INTERCOM\_CUSTOM\_DATA\_CLASSES setting. It is important to note that if you have the same dict key returned in more than one Custom Data Class the last class that is run (lower in the list) will overwrite the previous ones.

#### INTERCOM\_CUSTOM\_DATA\_CLASSES

Default = None

in settings.py:

```
INTERCOM_CUSTOM_DATA_CLASSES = [  
    'thepostman.utils.custom_data.IntercomCustomData',  
]
```



These are the settings that you can add to your django settings file to control django-intercom.

### 6.1 INTERCOM\_APPID

#### Required

Default: None

Example:

```
INTERCOM_APPID = "your appID"
```

### 6.2 INTERCOM\_SECURE\_KEY

#### Optional

Default: None

example:

```
INTERCOM_SECURE_KEY = "your security_code"
```

### 6.3 INTERCOM\_INCLUDE\_USERID

#### Optional

Whether to include the `user_id` in the intercom settings. Turn this off if you want to use the email as the key to identify a user.

Default: True

example:

```
INTERCOM_INCLUDE_USERID = False
```

## 6.4 INTERCOM\_ENABLE\_INBOX

**Optional**

Default: True

example:

```
INTERCOM_ENABLE_INBOX = True
```

## 6.5 INTERCOM\_ENABLE\_INBOX\_COUNTER

**Optional**

Default: True

example:

```
INTERCOM_ENABLE_INBOX_COUNTER = True
```

## 6.6 INTERCOM\_INBOX\_CSS\_SELECTOR

**Optional**

Default: '#Intercom'

example:

```
INTERCOM_INBOX_CSS_SELECTOR = '#Intercom'
```

## 6.7 INTERCOM\_CUSTOM\_DATA\_CLASSES

**Optional**

Default = None

example:

```
INTERCOM_CUSTOM_DATA_CLASSES = [  
    'thepostman.utils.custom_data.IntercomCustomData',  
]
```

## CHAPTER 7

---

### People who have worked on this project

---

- Ken Cochrane (@KenCochrane)
- Lee Longmore (@leelongmore)
- Neal Todd (@nealtodd)
- Danilo Barga (@dbrgn)
- Eduardo Raad (@eraad)
- Manuel Kaufmann (@humitos)



## CHAPTER 8

---

TODO

---

- Adding unit tests



## CHAPTER 9

---

### Indices and tables

---

- `genindex`
- `search`