
Django Mail Queue Documentation

Release 3.1.5

Derek Stegelman

Oct 14, 2018

Contents

1	Quick Start Guide	3
1.1	Requirements	3
1.2	Installation	3
1.3	Setup	3
1.4	Synchronously	4
1.5	Celery	4
1.6	Management Command/URL	4
1.7	Misc Settings	5
2	Usage	7
2.1	To Send an Email	7
2.2	Attaching Files	7
2.3	Adding a Reply To header	8
2.4	Sending to Multiple Recipients	8
2.5	Using the Management Command	8
3	Management Commands	9
3.1	Send Queued messages	9
3.2	Clear Sent messages	9
4	License	11
5	Contributing	13
6	Indices and tables	15

Mail Queue is a new way to send and keep track of email in your Django application. Mail queue stores each email you send and records success/failure. You can requeue up un-sent mail and review the mail generated by your app.

Contents:

As of 2.0 Django Mail Queue is now Python 3 compatible!

1.1 Requirements

Django Mail Queue requires:

```
python 2.7 or greater
django 1.8 or greater
```

Django Mail Queue is tested against Python 2.7, 3.x and Django 1.8 and 1.9.

If using Celery, you'll need celery 3 or greater.

1.2 Installation

Using pip:

```
pip install django-mail-queue
```

Go to <https://github.com/dstegelman/django-mail-queue> if you need to download a package or clone the repo.

1.3 Setup

Open `settings.py` and add `mailqueue` to your `INSTALLED_APPS`:

```
INSTALLED_APPS = (
    'mailqueue',
)
```

Mailqueue can be configured a few different ways:

- Configured to send mail synchronously in the web request.
- Configured to send mail asynchronously through Celery.
- Configured to queue up and send mail in bulk through a management command.
- Configured to queue up and send mail in bulk through a hitting a URL. (pending deprecation)

1.4 Synchronously

This is the default setting for mailqueue. You do not need to set any additional settings for this option.

1.5 Celery

Celery is disabled by default, you can turn it on the use of Celery and send emails in real time using `MAILQUEUE_CELERY` in settings:

```
MAILQUEUE_CELERY = True
```

Instead of using the cron job the celery task worker will attempt to send email when it's saved. The cron job will clean up any emails that get lost.

1.6 Management Command/URL

First, in order to queue up the mail and not send on `save()`, you'll need to set the queue up:

```
MAILQUEUE_QUEUE_UP = True
```

A cron job can be set up to work one of two ways: using a management command or an HTTP request. Both methods run the mail queue which grabs emails and sends them. To decrease load, it only tries to send 30 emails at a time. This number can be changed by using `MAILQUEUE_LIMIT` in settings:

```
MAILQUEUE_LIMIT = 50
```

Using the management command:

```
python manage.py send_queued_messages
```

You can also override `MAILQUEUE_LIMIT` by using the `--limit` or `-l` option:

```
python manage.py send_queued_messages --limit=10
```

HTTP request:

```
urlpatterns = patterns('',
    (r'^mail-queue/', include('mailqueue.urls')),
)
```

If you're running cron from another machine or can't run python directly, you can add the above to `urls.py` and use a utility like curl to hit `/mail-queue/`.

1.7 Misc Settings

You can force mail queue to use default file system storage with `MEDIA_ROOT` as the storage folder. You may want to do this because by default mail queue will use your default file storage, and attachments are known to not work against various storages such as S3 Boto.

To force Django's File System storage:

```
MAILQUEUE_STORAGE = True
```

To change the Attachment dir:

```
MAILQUEUE_ATTACHMENT_DIR = 'mailqueue-attachments'
```


2.1 To Send an Email

Create a new `MailerMessage()` object:

```
from mailqueue.models import MailerMessage

new_message = MailerMessage()
new_message.subject = "My Subject"
new_message.to_address = "someone@example.com"
new_message.cc_address = "carboncopy@yo.com"
new_message.bcc_address = "myblindcarboncopy@yo.com"
new_message.from_address = "hello@example.com"
new_message.content = "Mail content"
new_message.html_content = "<h1>Mail Content</h1>"
new_message.app = "Name of your App that is sending the email."
new_message.save()
```

When `save` is called, Django will immediately try to send the email. Should it fail, it will be marked as `unsent`, and wait for the next time the job is called. Of course, the `BCC` address is optional, as well as `html` content.

2.2 Attaching Files

File attachments can be added to the e-mail with `MailerMessage`'s `add_attachment` method:

```
from mailqueue.models import MailerMessage
from django.core.files import File

message = MailerMessage(to_address="foo@mail.com", from_address="bar@mail.com")

photo_one = File(open("Poznan_square.jpg", "rb"))
message.add_attachment(photo_one)
```

(continues on next page)

(continued from previous page)

```
# ...you can add more than one file attachment
photo_two = File(open("Poznan_Malta-lake.jpg", "rb"))
message.add_attachment(photo_two)

message.save()
```

2.3 Adding a Reply To header

You can add a reply to header to your emails by setting:

```
from mailqueue.models import MailerMessage

new_message = MailerMessage()
new_message.reply_to = 'reply@myawesomeaddress.com'
```

2.4 Sending to Multiple Recipients

To include more than one CC/BCC in your email, just separate the addresses with a comma:

```
message.cc_address = "one@mail.com, two@mail.com, three@mail.com"
message.bcc_address = "one@mail.com, two@mail.com, three@mail.com"
```

As of version 2.2.0 multiple recipients may be included in the *to_address* field as well:

```
message.to_address = "one@mail.com, two@mail.com, three@mail.com"
```

2.5 Using the Management Command

You can use the management command to send email:

```
python manage.py send_queued_messages --limit=20
```

Management Commands

3.1 Send Queued messages

You can use this management command to send email if you do not setup a cron job or use celery. You can specify a limit on the amount of emails you want to attempt to send at one time.:

```
python manage.py send_queued_messages 20
```

3.2 Clear Sent messages

You can use a management command to clear out successfully sent emails from the database:

```
./manage.py clear_sent_messages
```


CHAPTER 4

License

Copyright (c) 2011 - 2017 Derek Stegelman

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.

CHAPTER 5

Contributing

Mail Queue is hosted on github at <https://github.com/dstegelman/django-mail-queue>

Contributions are more than welcome!

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`