
django-mailrobot Documentation

Release 0.5.1

kaleissin

September 13, 2016

1	Installation	3
2	Usage	5
2.1	Address	5
2.2	Signature	5
2.3	MailBody	5
2.4	Mail	6
2.5	In code	6
3	Niceties	9
4	Testing	11
5	Models	13
6	Indices and tables	15

Stores and sends canned email responses.

Ever had to change the signature or add a recipient to N hardcoded emails spread all throughout your code? Hardcode no more! Use mailrobot instead.

Contents:

Installation

1. Install library, for instance with pip:

```
pip install django-mailrobot
```

2. Add library to your `INSTALLED_APPS` in your settings:

```
INSTALLED_APPS += ['mailrobot']
```

3. Add the tables.

Prior to django 1.7:

```
$ ./manage.py syncdb
```

With South:

```
$ ./manage.py schemamigration --initial mailrobot  
$ ./manage.py migrate mailrobot
```

Usage

Add mails and addresses through the django admin.

2.1 Address

Contains email-addresses.

```
addr1 = Address.objects.create(
    address='ceo@example.com',
    comment='Bossman'
)
```

This'll print as Bossman <ceo@example.com>. The comment is optional:

```
addr2 = Address.objects.create(
    address='ceo@example.com'
)
```

This'll print as ceo@example.com.

2.2 Signature

Contains signatures.

```
sig1 = Signature.objects.create(
    name='hello-world',
    sig="Test-system 'R us"
)
```

A good name makes the Signature easier to find in the admin.

The sig is a django template stored in a text-field and thus of unlimited length, but please do keep it to four lines or less!

2.3 MailBody

Contains the subject and (plaintext) body.

```
mb = MailBody.objects.create(
    name='hello-world',
    subject='Hello, {{ world|default:"World" }}!',
    body="""This is a test of the email-system."""
)
```

The name should be url-safe and is primarily used to find it again in the admin. A slugified version of the subject will work fine.

Both subject and body are django templates.

2.4 Mail

Combines a MailBody with an optional Signature and optional addresses.

```
m = Mail.objects.create(
    name='hello-world',
    signature=sig1,
    content=mb,
    sender=addr1,
    reply_to=addr2,
)
```

The name here is what you use to look up the mail in your code.

The sender and reply_to are optional and can be overridden later.

The recipients are Address querysets:

```
m.recipients = addresses_for_the_To_field
m.ccs = addresses_for_the_Cc_field
m.bccs = addresses_for_the_Bcc_field
```

When saved on the Mail like this they cannot be overridden later, only appended to.

2.5 In code

Fetch a mail-template:

```
template = Mail.objects.get(name='hello-world').
```

Fill it:

```
sender = 'Yep <overridden-from@example.com>'
recipients = ('extra1@example.com', u'Blåbærstetøy <extra2@example.com>')
context = {'world': 'Mailrobot'}

mail = template.make_message(
    sender=sender,
    recipients=recipients,
    context=context
)
```

The sender here is a **string** and overrides whatever is already stored in the mail. Notice that the recipients is an **iterable of strings**.

If you want to use one or more `Address` instead you need to convert the queryset to an iterable of strings, for instance via:

```
recipients = [str(address) for address in Address.objects.all()]
```

Have a look:

```
print mail.message
```

Send it:

```
mail.send()
```


Niceties

In case you need to send an email somewhere else for testing/debugging, clone an existing email in the admin:

1. Select it
2. Choose “Clone selected mails” in the action list
3. Hit “Go”

The clone will share everything with its original except the name, which will be suffixed with a timestamp.

Edit the name of the clone to what you need, change recipients, CCs, BCCs. Then, where you send the mail from, choose the clone if `DEBUG` is `True`.

Testing

Use `tox` if installed, else make `test APP=mailrobot`.

Models

Indices and tables

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

E

examine
 message, 7
examine generated email
 usage, 7

F

fetch
 template, 6
fetch template
 usage, 6
fill
 template, 6
fill template
 usage, 6

I

installation, 1
 INSTALLED_APPS, 1
 migrate, 1
 pip, 1
 syncdb, 1
INSTALLED_APPS
 installation, 1

M

message
 examine, 7
 send, 7
migrate
 installation, 1

P

pip
 installation, 1

S

send
 message, 7
send email

usage, 7

syncdb
 installation, 1

T

template
 fetch, 6
 fill, 6

U

usage, 3
 examine generated email, 7
 fetch template, 6
 fill template, 6
 send email, 7