
django-courier Documentation

Release 0.0.3

Maxime Haineault

Sep 27, 2017

Contents

1	Introduction	3
2	Installation	5
2.1	Django email settings example	5
2.2	Courier settings	5
3	Debugging	7
4	Indices and tables	9

django-courier allows easy configuration of notifications and management of email templates.

Contents:

CHAPTER 1

Introduction

For easier management of development and production environments I suggest to use a variable named *DEV* in your *settings.py*. This is technique optional, but the following examples will use it. Obviously, if *DEV* is True it means we are running it a development environment.

The key difference between *DEV* and *DEBUG* is that sometimes you need to set *DEBUG* to True in the production environment to troubleshoot a problem while *DEV* should *_never_* be True when in production.

The *settings.py* should contain something like this:

```
DEV = True
DEBUG = DEV
TEMPLATE_DEBUG = DEBUG
```


Just add `'courier'` in your `settings.INSTALLED_APPS` and sync your database.

Django email settings example

Courier will use standard Django configurations:

```
if DEV:
    EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'
    COURIER_FAIL_SILENTLY = False
else:
    EMAIL_HOST = 'smtp.example.com'
    EMAIL_PORT = 45
    EMAIL_HOST_USER = 'no-reply@example.com'
    EMAIL_HOST_PASSWORD = 'mypassword'
    EMAIL_USE_TLS = False
```

Courier settings

These settings are all optional.

Setting	Default	Description
<i>COURIER_FAIL_SILENTLY</i>	True	Whether or not to fail silently when sending emails
<i>COURIER_DEFAULT_EMAIL_FROM</i>	'no-reply@example.com'	Default "email from" to use (can be overridden for each EmailNotification)
<i>COURIER_DEFAULT_EMAIL_TO</i>	settings.ADMINS emails	Default "email to" to use (can be overridden for each EmailNotification)
<i>COURIER_IGNORED_CONTEXT_METHODS</i>	['add_to_class', 'copy_managers', 'mro']	Methods that should not show up in template variables helper (Warning: they are still available from the context)
<i>COURIER_EMAILTEMPLATE_ON_DELETE</i>	SET_NULL	Defines what happens when an email template linked to an email notification is deleted (can be <i>SET_DEFAULT</i> , <i>CASCADE</i> , <i>PROTECT</i> , <i>SET_NULL</i> or <i>DO_NOTHING</i> *)

* https://docs.djangoproject.com/en/dev/ref/models/fields/#django.db.models.ForeignKey.on_delete

To see debug output in your console when using the development server, just add a logger named *courier* in your *settings.py*.

Example:

```
LOGGING = {
    'version': 1,
    'disable_existing_loggers': False,
    'handlers': {
        'mail_admins': {
            'level': 'ERROR',
            'class': 'django.utils.log.AdminEmailHandler'
        },
        'console':{
            'level':'DEBUG',
            'class':'logging.StreamHandler',
        },
    },
    'loggers': {
        'django.request': {
            'handlers': ['mail_admins', 'console'],
            'level': 'ERROR',
            'propagate': True,
        },
        'courier': {
            'handlers': ['mail_admins', 'console'],
            'level': 'DEBUG',
            'propagate': True,
        },
    },
}
```


CHAPTER 4

Indices and tables

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