
django-email-hosts Documentation

Release 0.2

Feinheit AG

Feb 07, 2022

CONTENTS

1	Why?	3
2	Usage	5
3	EMAIL_HOSTS	7
4	email_hosts.backends.get_connection	9
5	Change log	11
5.1	Next version	11
5.2	0.2 (2022-02-07)	11
5.3	0.1 (2022-02-03)	11

WHY?

Some projects may want to send emails over multiple SMTP relays or smarthosts. Examples are sites running on multiple domains, e.g. `info@example.com` and `info@example.org`. It may be possible to use the same SMTP credentials for each sender address but if that isn't possible `django-email-hosts` may be a good solution for the problem.

USAGE

- **Install:** `pip install django-email-hosts`
- **Configure:** Add the `EMAIL_HOSTS` setting
- **Use:** Always explicitly use the SMTP connection returned by `email_hosts.backends.get_connection`

EMAIL_HOSTS

The keys of the `EMAIL_HOSTS` dictionary are defined by you and there's no deeper meaning to them. The values are DSNs inspired by [dj-email-url](#) DSNs.

An example configuration (which is possibly nonsensical) looks like this:

```
EMAIL_HOSTS = {
    "sendgrid": "submission://USER:PASSWORD@smtp.sendgrid.com?_default_from_
↳email=info@example.com",
    "mailgun": "submission://USER:PASSWORD@smtp.mailgun.com?_default_from_
↳email=info@example.org",
}
```

This configuration creates two SMTP backends, one using sendgrid and one using mailgun. The `_default_from_email` is completely optional. If the email message's `from_email` isn't set (resp. is equal to the `DEFAULT_FROM_EMAIL` setting) it automatically defaults to the per-backend value.

EMAIL_HOSTS.BACKENDS.GET_CONNECTION

The `get_connection` function expects a single key for the `EMAIL_HOSTS` setting above. Sending a single email using an explicit connection may look as follows, using the settings from above:

```
from django.core.mail import EmailMessage
from email_hosts.backends import get_connection

EmailMessage(
    "Hello",
    "World",
    to=["recipient@example.com"],
    connection=use_backend("sendgrid"),
).send()
```

`get_connection` currently silently returns the default email backend if the key doesn't exist in the `EMAIL_HOSTS` dictionary.

CHANGE LOG

5.1 Next version

5.2 0.2 (2022-02-07)

- Removed the `functools.cache` decorator from `get_connection` – constructing backends isn't that expensive and the danger of memory leaks when misusing the connection management is worse.
- Removed support for the `ssl_keyfile` and `ssl_certfile` parameters since they are never returned by `dj-email-url` anyway.
- Changed the implementation to use documented APIs instead of undocumented internals of Django.
- Switched from `dj-email-url` to `speckenv_django` for parsing the email URL.

5.3 0.1 (2022-02-03)

- Initial release!