django-email-bandit Documentation
Release 2.0

Caktus Consulting Group

Dec 17, 2020

Contents

Build Status

> **master**
>
> **develop**

A Django email backend for hijacking email sending in a staging/test environment. It extends the default SMTP backend to intercept outgoing emails and instead send them to a single email address that you specify. It does not intercept emails going to the site admins (as defined by the `ADMINS` setting) so it will not interfere with 500 error emails.

You can also configure the hijacked emails to be logged through standard Python logging. Mixin classes are provided to use the same hijack logic for any existing email backend such as django-ses.

CHAPTER 1

Requirements

- Python 3
- Django >= 2.2 (supported versions)

CHAPTER 2

Installation

To install django-email-bandit via pip:

```
pip install django-email-bandit
```

Add django-email-bandit to your installed apps:

```
INSTALLED_APPS = (
    ...
    'bandit',
    ...
)
```

For your test environment you should enable the backend:

```
EMAIL_BACKEND = 'bandit.backends.smtp.HijackSMTPBackend'
```

and set the email which will receive all of the emails:

```
BANDIT_EMAIL = 'bandit@example.com'
```

or even multiple addresses:

```
BANDIT_EMAIL = ['bandit@example.com', 'accomplice@example.com']
```

It's also possible to whitelist certain email addresses and domains:

```
BANDIT_WHITELIST = [
    'iloveemail@example.com',  # Just this specific email address
    'example.net'   # All email addresses @example.net
]
```

Documentation

Full project documentation is on Read the Docs.

Maintainer Information

We use Github Actions to lint (using pre-commit, black, isort, and flake8), test (using tox and tox-gh-actions), calculate coverage (using coverage), and build documentation (using sphinx).

We have a local script to do these actions locally, named `maintain.sh`:

```
$ ./maintain.sh
```

A Github Action workflow also builds and pushes a new package to PyPI whenever a new Release is created in Github. This uses a project-specific PyPI token, as described in the PyPI documentation here. That token has been saved in the `PYPI_PASSWORD` settings for this repo, but has not been saved anywhere else so if it is needed for any reason, the current one should be deleted and a new one generated.

As always, be sure to bump the version in `bandit/__init__.py` before creating a Release, so that the proper version gets pushed to PyPI.

Questions or Issues?

If you have questions, issues or requests for improvements please let us know on Github.

Development sponsored by Caktus Consulting Group, LLC.

Contents

6.1 Getting Started

django-email-bandit provides additional email backends for Django which reroute outgoing emails to another email address, typically a shared inbox or group email. Included in the intercepted message body is a list of the original recipients. This allows you to test functionality on your site which might otherwise send out emails to your users. Additionally, it allows recipient emails to be white-listed so that they are sent without modification to the original recipient(s). By default, django-email-bandit will not intercept emails to ADMINS, meaning that it will not interfere with the standard 500 error emails provided by Django.

6.1.1 Installing

The recommended method for installing django-email-bandit is using pip:

```
pip install django-email-bandit
```

The rest of this guide assumes that you have installed the latest stable version via pip.

6.1.2 Necessary Settings

Configure the following settings in your project settings.py. Add bandit to your INSTALLED_APPS:

```
INSTALLED_APPS = (
    # Other installed apps included here
    'bandit',
)
```

This is required to auto-discover the templates included with the application. To begin intercepting emails, change your EMAIL_BACKEND setting to one of the included backends. bandit.backends.smtp.HijackSMTPBackend is used to change the behavior of the default SMTPBackend backend:

```
EMAIL_BACKEND = 'bandit.backends.smtp.HijackSMTPBackend'
```

Finally you should set `BANDIT_EMAIL` to a valid email address where you will receive the emails which are intercepted:

```
BANDIT_EMAIL = 'bandit@example.com'
```

6.1.3 Customizing the Template

An optional step is to customize the templates used when an email is intercepted. The hijacked message body will contain a header defined by the `bandit/hijacked-email-header.txt` template. This template is given the context:

```
{
    'message': message,  # Original EmailMessage object,
    'previous_recipients': previous_recipients,  # List of the original recipients
    'previous_cc': previous_cc,  # List of the original CC'ed recipients
    'previous_bcc': previous_bcc,  # List of the original BCC'ed recipients
}
```

Note that the default header template doesn't include `previous_cc` or `previous_bcc`, so you'll need to create a custom template if you need those displayed.

6.2 Additional Email Backends

Beyond the `bandit.backends.smtp.HijackSMTPBackend` used in the getting started guide, django-email-bandit defines additional backends and contains helpers for hijacking already customized backends.

6.2.1 Using django-seacucumber

django-email-bandit supports sending email through SES via django-seacucumber.

To configure django-email-bandit, set your email backend as follows:

```
EMAIL_BACKEND = 'bandit.backends.seacuke.HijackSESBackend'
```

6.2.2 Only logging emails

In environments where your application may generate lots of emails all at once, it may be desirable to hijack emails to non-admins and have them logged, but not sent out. The logging is done using the standard Python logging facilities using the `bandit` logger at the `DEBUG` level.

django-email-bandit supports this with the `bandit.backends.smtp.LogOnlySMTPBackend` and the `bandit.backends.seacuke.LogOnlySESBackend`. For example, to configure django-email-bandit to only log emails to non-admins, but still send via SMTP emails to admins, configure your email backend like so:

```
EMAIL_BACKEND = 'bandit.backends.smtp.LogOnlySMTPBackend'
```

The log message which is generated can be customized by overriding the `bandit/hijacked-email-log-message.txt` template.

6.2.3 Hijacking Arbitrary Backends

You can also hijack email to an arbitrary Django email backend by wrapping a backend of your choice with the HijackBackendMixin. For example, if you wanted to send email through SES but prefer to use django-ses, you would create a class like this inside your project:

```python
from django_ses import SESBackend
from bandit.backends.base import HijackBackendMixin


class MyHijackBackend(HijackBackendMixin, SESBackend):
    """
    This backend intercepts outgoing messages drops them to a single email
    address, using the SESBackend in django-ses.
    """
    pass
```

and then set EMAIL_BACKEND as follows:

```python
EMAIL_BACKEND = 'my.project.path.to.MyHijackBackend'
```

Note that the order in which you specify the mixin and base class is very important. If you specify the mixin last, your email will not be hijacked.

6.3 Settings for django-email-bandit

These should be in your project settings.py.

6.3.1 BANDIT_EMAIL

The BANDIT_EMAIL defines the email address which recieves the hijacked emails. This defaults to the SERVER_EMAIL setting if not set:

```python
BANDIT_EMAIL = 'bandit@example.com'
```

This option can also be a list to send the hijacked email to multiple addresses:

```python
BANDIT_EMAIL = ['bandit@example.com', 'accomplice@example.com']
```

6.3.2 BANDIT_WHITELIST

By default django-email-bandit will intercept all outgoing emails unless they are sent to the SERVER_EMAIL or one of the emails in the ADMINS setting. If there are email addresses that should receive mail from your application normally, add those email addresses to BANDIT_WHITELIST. BANDIT_WHITELIST defaults to an empty tuple:

```python
BANDIT_WHITELIST = ('iloveemail@example.com', )
```

A domain can also be put in the BANDIT_WHITELIST. This will whitelist any email to that domain:

```python
BANDIT_WHITELIST = ('example.com', )
```

6.3.3 `BANDIT_REGEX_WHITELIST`

Similar to `BANDIT_WHITELIST`, but matches whole e-mail address by regular expressions:

```
BANDIT_REGEX_WHITELIST = ["ba.*it@bandit\\.com", "joe@.*\\.org"]
```

6.4 Django-Email-Bandit Changes

History of releases and changes to the django-email-bandit project. This project has reached a stable point in its development and new releases are primarily to ensure compatibility with newer versions of Django and Python.

6.4.1 Supported Django Versions

django-email-bandit aims to support the current security-supported releases of Django.

6.4.2 v2.0 (released 2020-12-17)

- Drop support for Python 2.7
- Drop support for Python 3.5
- Add support for Django 2.2, 3.0 and 3.1

6.4.3 v1.6 (released 2020-12-03)

- Add BANDIT_REGEXP_WHITELIST to allow match emails by regexps (#29)

6.4.4 v1.5 (released 2018-01-11)

- Added documentation regarding context variables available in the email template (#25)
- Allow BANDIT_EMAIL to be a list of addresses (#26)
- Add flake8 and coverage tox environments

6.4.5 v1.4 (released 2017-12-08)

- Added support for Django 2.0 and Python 3.5
- Dropped support for Django 1.5, 1.6, 1.7, 1.9 and Python 3.4
- Documentation improvements
- Support more email address formats (#19)

6.4.6 v1.3 (released 2017-10-31)

- Added support and test coverage for Django 1.11
- Added support and test coverage for Python 3.6
- Dropped support for Python 2.6 and Python 3.3.

6.4.7 v1.2 (Released 2016-11-08)

- Added support and test coverage for Django 1.10

6.4.8 v1.1 (Released 2016-01-20)

- Backends now intercept the CC and BCC fields as well
- Entire domains can be whitelisted. See the *settings documentation*.

6.4.9 v1.0 (Released 2014-03-21)

- Python 3 support
- Tox support for running tests with Travis CI integration
- Full project documentation
- Added `BANDIT_WHITELIST` setting to allow specified emails to pass through
- *Backwards incompatible* Dropped Django 1.2 support

6.4.10 v0.3.0

- Added mixin classes for logging hijacked emails rather than sending

6.4.11 v0.2.0

- Added support for hijacking django-seacucumber backend
- Added mixin to help with the creation of additional hijacked backends

6.4.12 v0.1.0

- Initial public release