
django-hipchat documentation

Release

Thread

April 02, 2014

1	Seamless HipChat integration for Django	3
1.1	Installation	3
1.2	Usage	3
1.3	Configuration	4
1.4	Links	5
	Python Module Index	7



Seamless HipChat integration for Django

This project provides easy-to-use integration between Django projects and the HipChat group chat and IM tool.

- Uses the templating system, rather than constructing messages “by hand” in your `views.py` and `models.py`.
- Easily enabled and disabled in certain environments, preventing DRY violations by centralising the logic to avoid sending messages in development or staging environments.
- Pluggable backend system for greater control over exactly how messages are transmitted to the HipChat API (eg. sent asynchronously using your queuing system)

1.1 Installation

1. Add `django_hipchat` to `INSTALLED_APPS`.
2. Ensure `django.template.loaders.app_directories.Loader` is in your `TEMPLATE_LOADERS`.

1.2 Usage

To send a message:

```
from django_hipchat import hipchat_message

hipchat_message('path/to/my_message.hipchat', {
    'foo': Foo.objects.get(pk=17),
})
```

Where `path/to/my_message.hipchat` (in your templates directory) might contain:

```
{% extends django_hipchat %}

{% block message %}
Message text here: {{ foo.bar|urlize }}
{% endblock %}

{% block room_id %}
Room name
{% endblock %}

{% block color %}
```

```
red
{% endblock %}
```

Required blocks:

- **message** – contains the message you wish to send. HTML entities are automatically escaped.

Required blocks which can be defaulted globally and overridden (see *Configuration*):

- **room_id** – ID or name of the room.
- **from** – Name the message will appear be sent from. Must be less than 15 characters long. May contain letters, numbers, -, _, and spaces.
- **auth_token** – Your HipChat authentication token (eg. 2aa7412fcb1b2e98067339603768c2)

Optional blocks:

- **color** – Background color for the message. One of “yellow”, “red”, “green”, “purple”, “gray”, or “random”. (default: yellow)

1.3 Configuration

1.3.1 HIPCHAT_BACKEND

Default: `"django_hipchat.backends.urllib"` (`"django_hipchat.backends.disabled"` if `settings.DEBUG`)

A string pointing to the eventual method that will actually send the message to the HipChat API. The default backend will send the message using the Python `urllib` library.

You can use this setting to globally disable sending messages to HipChat. You may need to set this to `django_hipchat.backends.disabled` when running tests or in your staging environment if you do not already set `DEBUG = True` in these environments.

If you are using a queue processor, you can wrap the supplied `urllib` backend so that messages are sent asynchronously and do not delay processing of requests:

```
from django_hipchat.backends import urllib as urllib_backend
from django_lightweight_queue.task import task
```

```
@task()
def queued_hipchat_backend(url, fail_silently):
    urllib_backend(url, fail_silently)
```

This would be enabled by setting `HIPCHAT_BACKEND` to (for example) `path.to.tasks.queued_hipchat_backend`.

You can also use the supplied `django_hipchat.backends.console` when developing. Instead of actually sending the message to HipChat, the console backend just writes the emails that would be sent to standard output.

1.3.2 HIPCHAT_AUTH_TOKEN

Default: None

Your HipChat authentication token. You can override on a per-message level by specifying a `{% block auth_token %}{% endblock %}` in your message templates.

1.3.3 HIPCHAT_MESSAGE_FROM

Default: None

Use this setting to set a default name the message will appear be sent from.

You can override on a per-message level by specifying a `{% block from %}{% endblock %}` in your message template.

1.3.4 HIPCHAT_MESSAGE_ROOM

Default: None

Use this setting to set a default ID or name of the room the message should appear in.

You can override on a per-message level by specifying a `{% block room_id %}{% endblock %}` in your message template.

1.3.5 HIPCHAT_FAIL_SILENTLY

Default: False

Whether errors should be silenced or raised to the user. As HipChat messages are often for administrators of a site and not the users, masking temporary errors with the HipChat API may be desired.

1.4 Links

Homepage/documentation: <https://django-hipchat.readthedocs.org/>

View/download code <https://github.com/thread/django-hipchat>

File a bug <https://github.com/thread/django-hipchat/issues>



Figure 1.1: See more open source projects from Thread.com

d

django_hipchat, 1