
Bambu Webhooks Documentation

Release 2.0

Steadman

May 03, 2014

Create webhooks and allow users to assign URLs to them

About Bambu Webhooks

This package allows web apps to provide third-party integration via webhooks. You as the developer can trigger a webhook by name, and provide an interface whereby the user can manage the URL to post the webhook's data to.

About Bambu Tools 2.0

This is part of a toolset called Bambu Tools. It's being moved from a namespace of `bambu` to its own 'root-level' package, along with all the other tools in the set. If you're upgrading from a version prior to 2.0, please make sure to update your code to use `bambu_webhooks` rather than `bambu.webhooks`.

Installation

Install the package via Pip:

```
pip install bambu-webhooks
```

Add it to your `INSTALLED_APPS` list:

```
INSTALLED_APPS = (  
    ...  
    'bambu_webhooks'  
)
```

Add `bambu_webhooks.urls` to your `URLconf`:

```
urlpatterns = patterns('',  
    ...  
    url(r'^webhooks/', include('bambu_webhooks.urls')),  
)
```

Run `manage.py syncdb` or `manage.py migrate` to setup the database tables.

Basic usage

Register a webhook within your models.py file.

```
from hashlib import md5
import bambu_webhooks

bambu_webhooks.site.register('webhook_name_',
    description = 'A description of the webhook'
)
```

In the `save()` method for your model, trigger any webhooks that have receivers attached, thus posting the data to the user's specified URL.

```
def save(self, *args, **kwargs):
    ...
    bambu_webhooks.send('webhook_name_', self.author,
        {
            'id': self.pk,
            'name': self.name
        },
        md5('testproject.myapp.mymodel:%d' % self.pk).hexdigest()
    )
```

Better with Bootstrap

This package, among most in the Bambu toolset is designed to work with [Bambu Bootstrap](#), a collection of flexible templates designed for web apps based on the Twitter Bootstrap framework. It's not a package requirement, but it'll mean the template structure and the context variables exposed by the view makes a little more sense.

Todo

- Allow webhooks to be categorised and/or filtered
- Prepare for internationalisation
- Write tests

Questions or suggestions?

Find me on Twitter (@iamsteadman) or visit my [blog](#).

7.1 Cron job

The Webhooks cron job takes all the temporarily-stored data meant to send to webhook receivers and sends that data. It runs every minute, allowing webhooks to work outside of the request-response cycle but without the need to setup an asynchronous handler like Celery.

Also, the use of the hash prevents the same webhook from being called multiple times for the same object, if that object rapidly changes inside a minute. This can happen if you put your webhook call inside the `save()` method of your model.

7.1.1 Installation

Once you've installed Bambu Webhooks, make sure to run `manage.py cron --setup` so that Bambu Cron can pick up the Webhooks cron job.

7.2 Menus

If you use [Bambu Navigation](#), the automatic menu builder for web apps, Bambu Webhooks will append a 'webhooks' item to the menu reserved for logged-in users. This links them through to the view that allows them to specify the URLs that are to be called when the webhook is fired.

7.3 Models

```
class bambu_webhooks.models.Action(*args, **kwargs)
```

A temporary place to store data that is to be sent to a webhook receiver outside of the request-response cycle

```
class bambu_webhooks.models.Receiver(*args, **kwargs)
```

A callback (or receiver) for a webhook call

Indices and tables

- *genindex*
- *modindex*
- *search*

b

`bambu_webhooks.models`, ??