Django GitHub webhooks

Jul 12, 2020

Contents:

CHAPTER 1

Table Of Contents

1.1 Installation

Install via pip:

```
$ pip install django-github-webhooks
```

Add app to *INSTALLED_APPS*:

```
INSTALLED_APPS = (
    ...
    "github_webhooks",
    ...
)
```

Add *SECRET* for *DJANGO_GITHUB_WEBHOOKS* in settings:

```
DJANGO_GITHUB_WEBHOOKS = {
    "SECRET": "secret-key"
}
```

Add URL patterns:

```
from github_webhooks import urls as github_webhooks_urls


urlpatterns = [
    ...
    url("webhooks/github/receive/", include(github_webhooks_urls)),
    ...
]
```

1.2 Configuration

1.2.1 Minimal configuration

```
DJANGO_GITHUB_WEBHOOKS = {
   "SECRET": "secret-key"
}
```

1.2.2 SECRET

Required: True

GitHub docs: https://developer.github.com/webhooks/creating/#secret

Setting a webhook secret allows you to ensure that *POST* requests sent to the payload URL are from GitHub. When you set a *SECRET*, you'll receive the *X-Hub-Signature* header in the webhook *POST* request.

You can also extend webhook view from *github_webhooks.views.GitHubWebhookView* to override *get_secret* method. More about that in *Secret key customization*.

1.2.3 ALLOWED_EVENTS

Required: False

Default:

```
DJANGO_GITHUB_WEBHOOKS = {
    ...
    "ALLOWED_EVENTS": [
        "issues",
    ],
    ...
}
```

GitHub docs: https://developer.github.com/webhooks/event-payloads/

ALLOWED_EVENTS is a list of all allowed events, that you want to handle.

1.3 User Guide

1.3.1 Using signals

On each allowed event github_webhooks executes a signal.

You can receive a signal by adding listeners like that:

```
# receivers.py
def issue_event(payload: dict, **kwargs) -> None:
    # Save / update etc. data
    pass
```

(continues on next page)

(continued from previous page)

```
# apps.py
from github_webhooks import signals


class MyApp(AppConfig):
    name = "myapp"

    def ready():
        from . import receivers
        signals.issues_signal.connect(receivers.issue_event)
```

1.4 Advanced User Guide

1.4.1 Secret key customization

You don't have to configure anything, except base setting, but if you want to store webhook secret key in the DB, or you have several webhooks, then you may need your own secret key retrieve implementation.

By default, secret key is obtained from *DJANGO_GITHUB_WEBHOOKS* settings, or rather, it happens in a view method, called *get_secret*.

```
class GitHubWebhookView(View):
    def get_secret(self) -> str:
        """
        Returns webhook's secret key.
        """
        secret = settings.DJANGO_GITHUB_WEBHOOKS.get("SECRET")
        if secret is None:
            raise ImproperlyConfigured("SECRET key for DJANGO_GITHUB_WEBHOOKS is not␣
↪specified!")
        else:
            return secret
```

For example, we want to handle multiple webhooks and retrieve secret key from DB by webhook id.

First of all, we need a model:

```
from django.db import models


class Webhook(models.Model):
    id = models.PositiveIntegerField(primary_key=True)
    secret = models.CharField(max_length=40)
```

Then, we need to override *github_webhooks.views.GitHubWebhookView*:

```
from github_webhooks.views import GitHubWebhookView
from .models import Webhook


class CustomWebhookView(GitHubWebhookView):
    def get_secret(self) -> str:
        return Webhook.objects.get(id=self.kwargs["id"]).secret
```

And also we need to override URLS:

```
from django.urls import path
from .views import CustomWebhookView

urlpatterns = [
    path("github/webhook/<int:id>/receive/", CustomWebhookView, name="github-webhook-
↪receive"),
]
```

That's it!

1.5 HISTORY

1.5.1 1.1.0 (2020/07/12)

- Added all webhook event signals.
- Binary payload for signals replaced with dictionary data.

1.5.2 1.0.0 (2020/06/14)

- Finished work on signature & headers validate.
- Added issues event signal.

1.5.3 0.1.0 (2020/06/14)

- Initial project setup.