Git Contributions

GitToken provides a Docker image and Dockerfile for configuring and listening to incoming GitHub contribution events via hypertext transfer protocol (HTTP) POST requests made by an organization's GitHub webhook service.

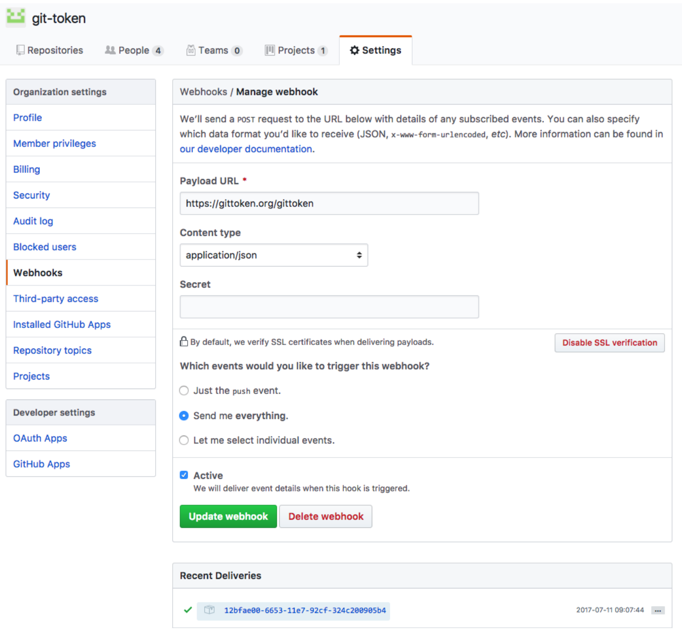
Request data is parsed and signed by the GitToken middleware handler, and sent to the GitToken contract to create and distribute tokens to contributors.

GitHub Webhook Events

Configuring a Webhook

> We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive. More information can be found in our developer documentation.

Under the **settings** tab in an organizations' GitHub dashboard, click **Webhook** on the left navigation section and add a new webhook.



Webhooks / **Manage webhook**

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, `x-www-form-urlencoded`, *etc*). More information can be found in our developer documentation.

Payload URL *

https://gittoken.org/gittoken

Content type

application/json

Secret

By default, we verify SSL certificates when delivering payloads.

Which events would you like to trigger this webhook?

○ Just the `push` event.

● Send me **everything**.

○ Let me select individual events.

☑ **Active**
We will deliver event details when this hook is triggered.

Recent Deliveries

✓ `12bfae00-6653-11e7-92cf-324c200905b4` 2017-07-11 09:07:44

GitToken, by default, sets the webhook endpoint to be `/gittoken`. This endpoint is customizable in the configuration file of the GitToken Docker service. [1]

Enter the url of the organization's GitHub webhook endpoint in the `payload URL` field of the webhook settings page. This is the endpoint that will receive POST requests when a contribution is made to any of an organizations' repositories.

Ping Event

The `ping` event is the first event sent by the GitHub webhook service. Its purpose is to test the endpoint configuration and establish the keystore and contract for the GitToken server.

The webhook service will display an alert if the endpoint responds with a `200` success status. A successful response will include JSON data about the keystore account, contract creation transaction receipt, and current details about the blockchain the GitToken server is connected to.

```
{
  "accounts": { ... },
  "contract": { ... },
  "blockchain": { ... }
}
```

Otherwise, the webhook service will display an error message with either a `400` or `500` error status.

Upon receiving a ping event, the GitToken server checks if a keystore and GitToken contract already exist. If either of those do not exist, the GitToken server attempts to create the keystore and deploy a contract with the configured parameters provided to the GitToken server instance.

Common reasons a contract may not deploy may be due to inadequate funds of the Ethereum account tied to the keystore. A faucet is provided in the testnet and beta launch to provide the minimum amount necessary to create a contract.

The server will respond with an error message if the contract could not be created.

---

[1] e.g., the webhook endpoint for GitToken's repositories is `https://GitToken.org/gittoken` ↩