

- [Introduction](#)
- [GitHub Web Hook Events](#)
- [GitTokens](#)
- [Contracts](#)
  - [GitToken.sol](#)
  - [GitTokenLib.sol](#)
- [Token Auctioneering](#)
  - [Evaluation of Open Source Projects](#)
- [Authentication](#)
- [References](#)

# Introduction

---

Over the past 9 years GitHub has quickly become the largest host of source code in the world, managing nearly 57 million repositories and 100 million pull requests for over 26 million users. (GitHub et. al. 2017) As the open source movement continues to gain momentum so to does the number of individual contributors to these repositories. In some cases these repositories had over 10,000 contributors. (GitHub et. al. 2016)

At the same time, the emergence of cryptographic networks and assets, such as Ethereum, has created new protocols for sending and managing value. Ethereum is a cryptographic network for running distributed programs; allowing users of Ethereum to send peer-to-peer (P2P) transactions and interact with smart contracts deployed on the network.

Ethereum smart contracts are software applications written in a high-level scripting language and compiled into byte code to be run on a version of the Ethereum Virtual Machine (EVM). The EVM interprets the byte code instruction set and translates the the program into machine code to be executed. (Buterin et. al. 2017)

GitToken combines the work flows of Git version control system leveraged by GitHub's web-based source code management platform and the Ethereum network to provide a set of open-source software tools and programs to enable any GitHub user to issue their own ERC20 tokens to incentivize and reward contributors, and monitor the fundamentals of their projects by integrating token generation with git contributions.

Contributions are mapped to GitHub web hook events, and include but are not limited to, creating issues, committing code, merging branches, forking repositories, and reviewing code. GitToken provides a Docker container to deploy a server for configuring and listening to GitHub web hook events.

Contributors receive tokens through interacting with an organization that has configured a GitToken server and has setup a GitHub web hook. When a contributor creates a new event, her/his GitHub login username is provided in the web hook request. Each username is mapped to an Ethereum address in the GitToken contract.

Contributors verify their GitHub identity by authenticating into GitHub using their Open Authorization (OAuth) token credentials. Contributors authenticate themselves with the token contract using the GitToken server authentication URL associated with an organization. For example, the authentication URL for the GitToken GTK token contract is `https://GitToken.org/auth/github` . If a contributor has not yet verified their identity with the contract, their contribution rewards will be held by the contract until their identity has been verified.

The Ethereum ecosystem has adopted a de facto contract interface for transacting value on top of the Ethereum network, the ERC20 protocol. The ERC20 protocol allows tokens to be exchanged over-the-counter (OTC) with private parties using Ethereum contracts. While the standard is still evolving, many developers have used the ERC20 token to represent utility or rights in their projects and have offered tokens to the public to raise funding for open source development. (Aitken 2017)

GitToken will offer GTK tokens to represent contributions made to the organization's GitHub [repositories](#). A portion of tokens issued are automatically auctioned to bidders by the GitToken contract upon each event.

---

## Git Contributions

---

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

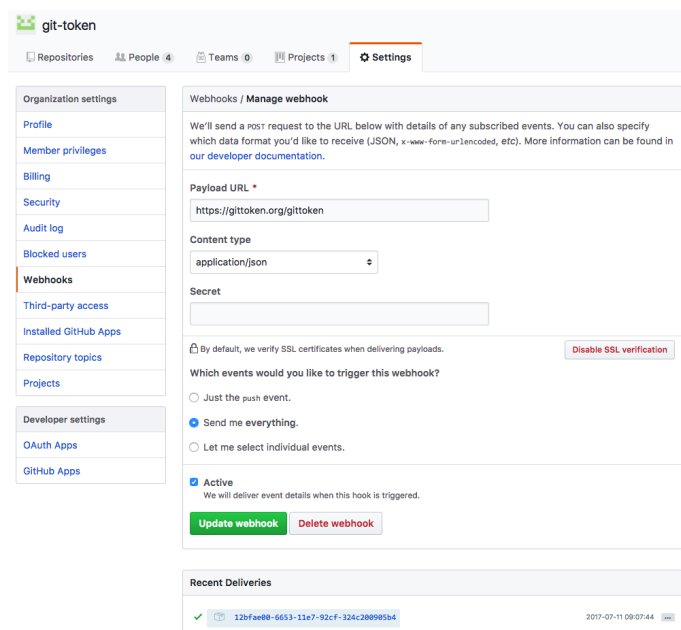
Request data is parsed and signed by the GitToken middleware handler, and sent to the GitToken contract to create and distribute tokens to contributors.

# 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 organization's GitHub dashboard, click **Webhook** on the left navigation section and add a new webhook.



The screenshot shows the GitHub 'Manage webhook' settings page for an organization named 'git-token'. The left sidebar contains navigation links for Organization settings (Profile, Member privileges, Billing, Security, Audit log, Blocked users, Webhooks, Third-party access, Installed GitHub Apps, Repository topics, Projects) and Developer settings (OAuth Apps, GitHub Apps). The 'Webhooks' link is highlighted. The main content area is titled 'Webhooks / Manage webhook' and includes a descriptive paragraph about sending POST requests. It features input fields for 'Payload URL' (set to 'https://gittoken.org/gittoken'), 'Content type' (set to 'application/json'), and 'Secret'. There are checkboxes for 'By default, we verify SSL certificates when delivering payloads.' (checked) and 'Disable SSL verification'. Below this, there are radio buttons for 'Which events would you like to trigger this webhook?' with options: 'Just the push event.', 'Send me everything.' (selected), and 'Let me select individual events.'. At the bottom, there is an 'Active' checkbox (checked) and a note 'We will deliver event details when this hook is triggered.'. Two buttons, 'Update webhook' and 'Delete webhook', are at the bottom of the form. A 'Recent Deliveries' section at the very bottom shows a single successful delivery with a green checkmark, a long alphanumeric ID, and a timestamp '2017-07-11 08:07:44'.

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

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 does not exist, the GitToken server attempts to create the keystore and deploy a contract with the configured parameters provided to the GitToken server instance.

One common reason a contract may not deploy may be due to inadequate funds in the Ethereum account tied to the keystore. For the purpose of the GitToken alpha on the Ropsten testnet a faucet can be used 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.

## Event Types

# GitTokens

---

## Contracts

---

[GitToken.sol](#)

[GitTokenLib.sol](#)

# Token Auctioneering

---

Ethereum ERC20 tokens have recently provided a new mechanism for funding projects. While tokens have brought liquidity to start-ups, it has also brought mis-pricing and speculation.

Many of the projects that have offered tokens to build their projects have used Git and GitHub to manage and develop software collaboratively, yet independently.

There is a clear relationship between the number of git contributions a project has and the market capitalization of the project.

## Evaluation of Open Source Projects

```
TESTER = document.getElementById('test')
Plotly.plot(TESTER, [{
  x: [38438497236, 18401040701, 1376206596, 235578951, 228864420, 87398581, 5623927],
  y: [14409, 8680, 8299, 3958, 258, 1492, 706],
  name: 'Contributions',
  text: ['Bitcoin', 'Ethereum', 'Ethereum Classic', 'Golem', 'Gnosis', 'Status', 'Aragon'],
  textposition: 'top center',
  marker: {
    size: [38.43, 18.40, 1.37, .23, .22, .08, .05]
  },
  mode: 'markers+text'
}], {
  height: 600,
  margin: { t: 100, l: 50 },
  title: 'Git Commits Vs. Market Capitalization'
});
```

GitToken maps the total supply of the token to contributions made to the project.

## Authentication

---

A contributor verifies their identity by associating an Ethereum address to their OAuth GitHub credentials using the web application user interface.

## References

---

Aitken. 2017. "Gnosis' Prediction Market Scores \$12.5M in 'Record-Breaking' Crypto Auction." *<https://www.forbes.com/Sites/Rogeraitken/2017/04/24/Gnosis-Prediction-Market-Scores-12-5m-in-Record-Breaking-Crypto-Auction/#3afec93ce87d>*. Accessed July 11.

Buterin et. al. 2017. "A Next-Generation Smart Contract and Decentralized Application Platform." *<https://github.com/Ethereum/Wiki/wiki/White-Paper>*. Accessed July 12.

GitHub et. al. 2017. "Celebrating Nine Years of GitHub with an Anniversary Sale." *<https://github.com/Blog/2345-Celebrating-Nine-Years-of-Github-with-an-Anniversary-Sale>*. Accessed July 12.

— — —. 2016. "The State of the Octoverse 2016." *<https://octoverse.github.com/>*. Accessed December 31.

---

1. e.g., the webhook endpoint for [GitToken](#)'s repositories is

`https://GitToken.org/gittoken` ↩