adnanh / **webhook**

webhook is a lightweight incoming webhook server to run shell commands

#webhook #go #devops #sysadmin #deploy #redeploy #github #bitbucket #shell #web #server #incoming #execute #http #https #hook #hooks #webhooks #integration #ci

327 commits | **2** branches | **26** releases | **28** contributors | MIT

Branch: master ▾ | New pull request | Create new file | Upload files | Find file | Clone or download ▾

adnanh Merge pull request #281 from ruliezz/patch-1 … Latest commit 385898b on Nov 16

.github	Added .github/ISSUE_TEMPLATE.md (optional)	a year ago
Godeps	Vendor	a year ago
docs	Updated with a Gitea example	a month ago
hook	Merge pull request #206 from dcj/feature/incoming-payload-content-type	3 months ago
test	Merge pull request #228 from moorereason/iss225	10 months ago
vendor	Vendor	a year ago
.gitignore	Allow hookecho to exit with codes other than 0	a year ago
.travis.yml	Add Travis CI and Appveyor configurations	10 months ago
CONTRIBUTING.md	Added CONTRIBUTING.md (optional)	a year ago
LICENSE	new path	4 years ago
Makefile	Fix Makefile to include .exe extension for windows builds	a month ago
README.md	Update README.md	a month ago
appveyor.yml	Add Travis CI and Appveyor configurations	10 months ago
hooks.json.example	add support for setting global response headers using -header flag	3 years ago
hooks.json.tmpl.example	Allow hooks file to be parsed as a template	a year ago
hooks.yaml.example	Use the github.com/ghodss/yaml library to unserialize webhooks.	2 years ago
hooks.yaml.tmpl.example	Allow hooks file to be parsed as a template	a year ago
signals.go	Started work on multi file hooks loading	2 years ago
signals_windows.go	Refactor signal handling and Windows support	3 years ago
webhook.go	Bump version to 2.6.9	a month ago
webhook_test.go	Fix another race condition in TestWebhook	10 months ago

README.md

What is webhook?

webhook is a lightweight configurable tool written in Go, that allows you to easily create HTTP endpoints (hooks) on your server, which you can use to execute configured commands. You can also pass data from the HTTP request (such as headers, payload or query variables) to your commands. webhook also allows you to specify rules which have to be satisfied in order for the hook to be triggered.

For example, if you're using Github or Bitbucket, you can use webhook to set up a hook that runs a redeploy script for your project on your staging server, whenever you push changes to the master branch of your project.

If you use Mattermost or Slack, you can set up an "Outgoing webhook integration" or "Slash command" to run various commands on your server, which can then report back directly to you or your channels using the "Incoming webhook integrations", or the appropriate response body.

webhook aims to do nothing more than it should do, and that is:

1. receive the request,
2. parse the headers, payload and query variables,

3. check if the specified rules for the hook are satisfied,
4. and finally, pass the specified arguments to the specified command via command line arguments or via environment variables.

Everything else is the responsibility of the command's author.

Hookdoo

If you don't have time to waste configuring, hosting, debugging and maintaining your webhook instance, we offer a **SaaS** solution that has all of the capabilities webhook provides, plus a lot more, and all that packaged in a nice friendly web interface. If you are interested, find out more at hookdoo website. If you have any questions, you can contact us at info@hookdoo.com

Getting started

Installation

Building from source

To get started, first make sure you've properly set up your Golang environment and then run the

```
$ go get github.com/adnanh/webhook
```

to get the latest version of the webhook.

Using package manager

Ubuntu

If you are using Ubuntu linux (17.04 or later), you can install webhook using `sudo apt-get install webhook` which will install community packaged version.

Debian

If you are using Debian linux ("stretch" or later), you can install webhook using `sudo apt-get install webhook` which will install community packaged version (thanks @freeekanayaka) from https://packages.debian.org/sid/webhook

Download prebuilt binaries

Prebuilt binaries for different architectures are available at GitHub Releases.

Configuration

Next step is to define some hooks you want webhook to serve. Begin by creating an empty file named `hooks.json`. This file will contain an array of hooks the webhook will serve. Check Hook definition page to see the detailed description of what properties a hook can contain, and how to use them.

Let's define a simple hook named `redeploy-webhook` that will run a redeploy script located in `/var/scripts/redeploy.sh`. Make sure that your bash script has `#!/bin/sh` shebang on top.

Our `hooks.json` file will now look like this:

```
[
  {
    "id": "redeploy-webhook",
    "execute-command": "/var/scripts/redeploy.sh",
    "command-working-directory": "/var/webhook"
  }
]
```

You can now run webhook using

```
$ /path/to/webhook -hooks hooks.json -verbose
```

It will start up on default port 9000 and will provide you with one HTTP endpoint

```
http://yourserver:9000/hooks/redeploy-webhook
```

Check webhook parameters page to see how to override the ip, port and other settings such as hook hotreload, verbose output, etc, when starting the webhook.

By performing a simple HTTP GET or POST request to that endpoint, your specified redeploy script would be executed. Neat!

However, hook defined like that could pose a security threat to your system, because anyone who knows your endpoint, can send a request and execute your command. To prevent that, you can use the `"trigger-rule"` property for your hook, to specify the exact circumstances under which the hook would be triggered. For example, you can use them to add a secret that you must supply as a parameter in order to successfully trigger the hook. Please check out the Hook rules page for detailed list of available rules and their usage.

Templates

webhook can parse the `hooks.json` input file as a Go template when given the `-template` CLI parameter. See the Templates page for more details on template usage.

Using HTTPS

webhook by default serves hooks using http. If you want webhook to serve secure content using https, you can use the `-secure` flag while starting webhook. Files containing a certificate and matching private key for the server must be provided using the `-cert /path/to/cert.pem` and `-key /path/to/key.pem` flags. If the certificate is signed by a certificate authority, the cert file should be the concatenation of the server's certificate followed by the CA's certificate.

CORS Headers

If you want to set CORS headers, you can use the `-header name=value` flag while starting webhook to set the appropriate CORS headers that will be returned with each response.

Interested in running webhook inside of a Docker container?

You can use almir/webhook docker image, or create your own (please read this discussion).

Examples

Check out Hook examples page for more complex examples of hooks.

Guides featuring webhook

- Webhook & JIRA by @perfecto25
- Trigger Ansible AWX job runs on SCM (e.g. git) commit by @jpmens
- Deploy using GitHub webhooks by @awea
- Setting up Automatic Deployment and Builds Using Webhooks by Will Browning
- Auto deploy your Node.js app on push to GitHub in 3 simple steps by Karolis Rusenas
- Automate Static Site Deployments with Salt, Git, and Webhooks by Linode
- Using Prometheus to Automatically Scale WebLogic Clusters on Kubernetes by Marina Kogan
- Github Pages and Jekyll - A New Platform for LACNIC Labs by Carlos Martínez Cagnazzo
- How to Deploy React Apps Using Webhooks and Integrating Slack on Ubuntu by Arslan Ud Din Shafiq
- Private webhooks by Thomas
- Adventures in webhooks by Drake
- GitHub pro tips by Spencer Lyon
- XiaoMi Vacuum + Amazon Button = Dash Cleaning by c0mmensal
- VIDEO: Gitlab CI/CD configuration using Docker and adnanh/webhook to deploy on VPS - Tutorial #1 by Yes! Let's Learn Software Engineering

- ...
- Want to add your own? Open an Issue or create a PR :-)

Community Contributions

See the webhook-contrib repository for a collections of tools and helpers related to webhook that have been contributed by the webhook community.

Need help?

Check out existing issues to see if someone else also had the same problem, or open a new one.

Support active development

Sponsors

DigitalOcean

DigitalOcean is a simple and robust cloud computing platform, designed for developers.

BrowserStack

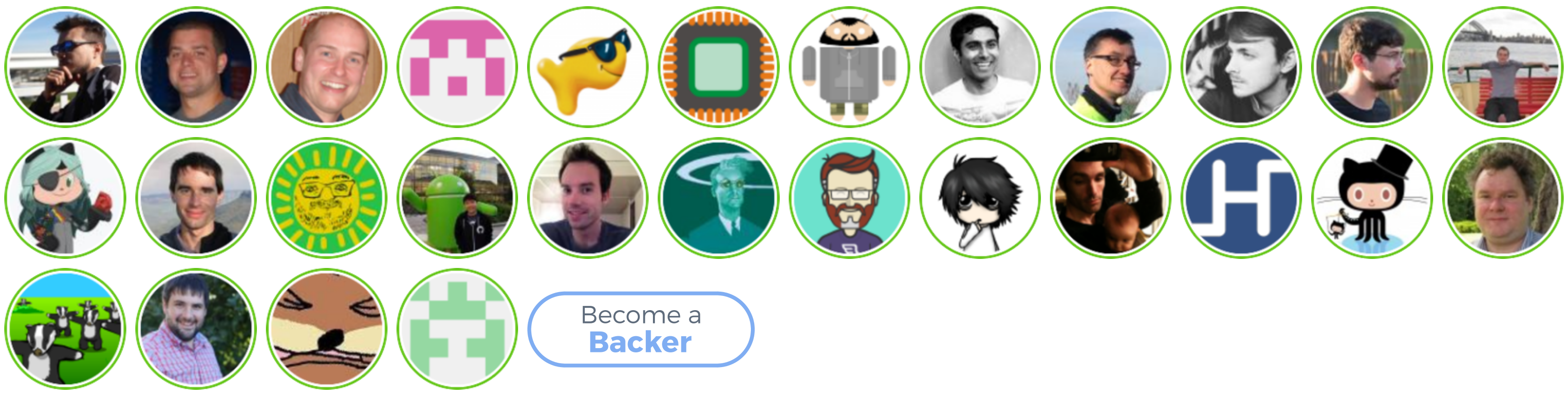
BrowserStack is a cloud-based cross-browser testing tool that enables developers to test their websites across various browsers on different operating systems and mobile devices, without requiring users to install virtual machines, devices or emulators.

Support this project by becoming a sponsor. Your logo will show up here with a link to your website.

Become a Sponsor

By contributing

This project exists thanks to all the people who contribute. Contribute!.



Become a Backer

By giving money

- OpenCollective Backer
- OpenCollective Sponsor
- PayPal
- Patreon
- Faircode
- Flattr

Thank you to all our backers!

Become a Backer

License