

Forum:[Hack Forums](#)

Thread: [Hooks using Git](#)

Remix - September 10, 2015, 11:21 am

Introduction

Like many other VCS Git offers a method to execute custom scripts when a pre triggered action occurs.

Git offers two groups of hooks: client side and server side hooks.

The client side hooks are pre triggered by actions like committing, merching, pushing etc while server side hooks are triggered by actions like receiving pushed commits etc.

You can use these triggers for whatever you like.

Client side hooks

All of the hooks are stored in .git/hooks

When you initialize new repositories Git generates the hooks with a bunch of example scripts.

They are actually already useful but you can modify them.

These example scripts are written as shell scripts with some Perl.

But basicly every executable script language would work.

For example you could use Ruby or Python.

If you want to use the pre generated scripts you just have to delete their extension.

If you want to create your own scripts you just have to name the new file like the example file for the specific trigger.

If you've removed the extension or created a new file it should be called everytime the action occurs.

Here's a little example for a commit message:

For these purpose I create a new file named commit-msg in the .git/hooks directory of my repository. (You could also just empty the existing file and remove the file extension)

The file contains this little code:

Code `#!/bin/sh`

`echo "# test message #"`

If you commit changes now it will look like this:

[img]<http://hornyhf.com/hcQQ.png>[/img]

You can customize your script how you want.

Note: the parameters are stored in the default command line parameter variables

Server side hooks

You can use custom scripts just if you're the system administrator.
So you can't use them for remote hosts like GitHub.

Server side hooks run before and after pushing to the server.
The hook can return an error message to the client or exit non-zero.
You can set up a push policy how ever you want.
On the server side you have the triggers pre-receive, update and post-receive.
I think the names should explain when they are triggered.
They are located in the same directory like the client side hooks (just on the server) and you can edit them on the same way.

Closing words

I know that wasn't a long tutorial but it's a really nice and useful technique which I wanted to share with the community.
I hope you enjoyed this little guide.

Greetings Remix