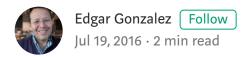


Pre-commit git hook that prevents commits with undesired words (usually debugging statements)



Git hooks allow you to customize Git. From the Git documentation:

Git has a way to fire off custom scripts when certain important actions occur. There are two groups of these hooks: client-side and server-side. Client-side hooks are triggered by operations such as committing and merging, while server-side hooks run on network operations such as receiving pushed commits. You can use these hooks for all sorts of reasons.

One of these hooks is the *pre-commit* hook.

The pre-commit hook is run first, before you even type in a commit message. It's used to inspect the snapshot that's about to be committed, to see if you've forgotten something, to make sure tests run,

# or to examine whatever you need to inspect in the code

I'm using the following script as a *pre-commit* hook to verify if a list of "undesired" words are present in the files I'm intended to commit such console output, debugging information or keys/tokens/passwords:

```
# This script verifies if a list of "undesired" words are presented in the files you are intende
     # output, debugging information or keys/tokens/passwords
     # Based on the git hook created by Mark Story
     # http://mark-story.com/posts/view/using-git-commit-hooks-to-prevent-stupid-mistakes
     # Instructions:
     # Put this file into your .git/hooks folder and set as executable (chmod +x pre-commit)
10
11
12
     # If you want to skip the hook just add the --no-verify: git commit --no-verify
13
14
15
16
     #!/bin/sh
17
     # Modify this
     # LIST='list\|of\|words\|splitted\|by\|slash\|and\|pipe'
     LIST="puts\|debugger\|binding.pry\|alert(\|console.log("
20
```

```
22
     if git rev-parse --verify HEAD >/dev/null 2>&1; then
         against=HEAD
23
24
     else
         against=4b825dc642cb6eb9a060e54bf8d69288fbee4904
25
     fi
26
27
     for FILE in `git diff-index --name-status --cached $against -- | cut -c3-`; do
28
         # Check if the file contains one of the words in LIST
         if grep -w $LIST $FILE; then
30
           echo $FILE." has one of the word you don't want to commit. Please remove it"
31
           exit 1
32
33
         fi
34
           done
     exit
nre-commit hosted with O by GitHub
                                                                                              view raw
```

## To install the hook you need to save this file as

.git/hooks/pre-commit

#### and set it as executable

chmod +x .git/hooks/pre-commit

If you want to skip the hook just add the — *no-verify* flag:

```
git commit --no-verify
```

I use git aliases to make this easy:

```
# .zshrc or .bash_aliases
alias gc='git commit'
alias gcv='git commit --no-verify'
```

(this post was originally published in my tumblr: http://edgar.tumblr.com/post/89077107659/pre-commit-git-hook-that-prevents-commits-with)

```
Git Hooks Pre Commit
```

#### **Discover Medium**

Welcome to a place where words matter. On Medium, smart voices and original

### **Make Medium yours**

Follow all the topics you care about, and we'll deliver the best stories for you to your homepage and inbox. Explore

## Become a member

Get unlimited access to the best stories on Medium — and support writers while you're at it. Just \$5/month. Upgrade ideas take center stage - with no ads in sight. Watch

About Help Legal