

Git recipe book

Edward McGuire

April 25, 2023

Abstract

Here I keep notes to myself about how to use Git.

Copyright © 2023 Edward McGuire.

Contents

1 Write a good commit message	1
2 Put one's own project under version control	2
3 Enter new, changed, or removed files in a commit event, and commit	2
4 Test before commit	2
5 Review changes	3
6 Revert changes	3
7 Fix the last commit message	4

1 Write a good commit message

A well-formed commit message has the following properties.

The first line is the subject. Write it in the imperative mood, using present tense. Avoid subjects longer than about 50 characters. For example:

```
$ git commit -m 'add syslog support'
```

Optionally, add explanatory body text. Separate the body text from the subject with a blank line. Wrap body text to under 80 columns. For example:

```
$ git commit
```

This will open your editor so you can enter a multiline message.

See: <https://commit.style/>

See: <https://gist.github.com/ericavonb/3c79e5035567c8ef3267>

2 Put one's own project under version control

```
$ cd ~/src/(myproject)
$ git init
$ ls .git
$ git add (files to include in initial commit)
$ git commit -m 'put initial files in revision control'
```

See: <https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository>

3 Enter new, changed, or removed files in a commit event, and commit

```
$ touch filea
$ git add filea
$ vi fileb
$ git add fileb
$ rm filec
$ git rm filec
$ git status -s
$ git commit -m 'add filea, modify fileb, drop filec'
```

See: <https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository>

4 Test before commit

Create an executable script `.git/hooks/pre-commit`. Its exit status will tell `git` whether the test passed. An exit status of 0 will allow the commit to proceed. An exit status of non-zero will abort the commit.

```
2 <pre-commit 2>≡
  #! /bin/sh
  # in .git/hooks:
  # pre-commit -> ../../pre-commit
  make test
```

This code is written to file `pre-commit`.

```

3 <Makefile 3>≡
  all :: pre-commit
  all :: .git/hooks/pre-commit
  all :: git.pdf

  install :: all

  .git/hooks/pre-commit : ; cd .git/hooks ; ln -sf ../../pre-commit .

  Makefile : git.nw.sentinel ;
  pre-commit : git.nw.sentinel ; chmod 755 $@
  git.nw.sentinel : git.nw ; noweb -t $< && touch $@

  test :: all
  test :: git.pdf ; xpdf git.pdf

  install :: /var/www/metaed.com/root/papers/git.pdf
  /var/www/metaed.com/root/papers/git.pdf : git.pdf ; cp $< $@
  git.pdf : git.tex ; latexmk -pdf git.tex
  git.tex : git.nw ; noweb -o git.nw

  clean :: ; rm -f *~ .*~
  clean :: ; rm -f git.tex git.pdf
  clean :: ; rm -f git.aux git.fdb_latexmk git.fls git.log git.out git.toc
  clean :: ; rm -f Makefile
  clean :: ; rm -f pre-commit
  clean :: ; rm -f git.nw.sentinel

```

This code is written to file `Makefile`.

5 Review changes

List commits:

```
$ git log
```

List commits with diffs (patches):

```
$ git log -p
```

6 Revert changes

Before the file is staged:

```
$ git restore FILE
```

After the file is staged:

```
$ git restore --staged FILE
```

The `git status` command types these instructions also.

See: <https://git-scm.com/book/en/v2/Git-Basics-Undoing-Things>

7 Fix the last commit message

```
$ git commit --amend
```