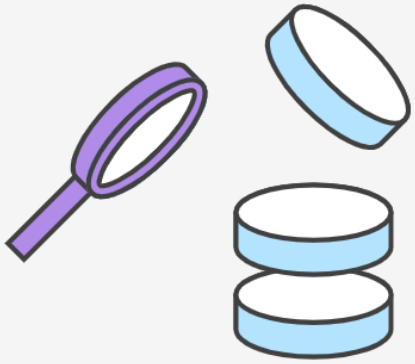


Git blame

git status / git tag / git blame

The `git blame` command is a versatile troubleshooting utility that has extensive usage options. The high-level function of `git blame` is the display of author metadata attached to specific committed lines in a file. This is used to examine specific points of a file's history and get context as to who the last author was that modified the line. This is used to explore the history of specific code and answer questions about what, how, and why the code was added to a repository.

`Git blame` is often used with a GUI display. Online Git hosting sites like Bitbucket offer *blame views* which are UI wrappers to `git blame`. These views are referenced in collaborative discussions around pull requests and commits. Additionally, most IDE's that have Git integration also have dynamic blame views.

How It Works

In order to demonstrate `git blame` we need a repository with some history. We will use the open source project git-blame-example. This open source project is a simple repository that contains a README.md file which has a few commits from different authors. The first step of our `git blame` usage example is to `git clone` the example repository.

```
git clone https://kevzettler@bitbucket.org/kevzettler/
```

Now that we have a copy of the example code we can start exploring it with `git blame`. The state of the example repo can be examined using `git log`. The commit history should look like the following:

```
$ git log
commit 548dabed82e4e5f3734c219d5a742b1c259926b2
Author: Juni Mukherjee <jmukherjee@atlassian.com>
Date: Thu Mar 1 19:55:15 2018 +0000
Another commit to help git blame track the who, the wh
commit eb06faedb1fdd159d62e4438fc8dbe9c9fe0728b
Author: Juni Mukherjee <jmukherjee@atlassian.com>
Date: Thu Mar 1 19:53:23 2018 +0000
Creating the third commit, along with Kev and Albert,
commit 990c2b6a84464fee153253dbf02e845a4db372bb
Merge: 82496ea 89feb84
Author: Albert So <aso@atlassian.com>
Date: Thu Mar 1 05:33:01 2018 +0000
Merged in albert-so/git-blame-example/albert-so/readme
README.md edited online with Bitbucket
commit 89feb84d885fe33d1182f2112885c2a64a4206ec
Author: Albert So <aso@atlassian.com>
Date: Thu Mar 1 00:54:03 2018 +0000
README.md edited online with Bitbucket
```

`git blame` only operates on individual files. A file-path is required for any useful output. The default execution of `git blame` will simply output the commands help menu. For this example, we will operate on the README.MD file. It is a common open source software practice to include a README file in the root of a git repository as documentation source for the project.

```
git blame README.MD
```

Executing the above command will give us our first sample of blame output. The following output is a subset of the full blame output of the README. Additionally, this output is static is reflective of the state of the repo at the time of this writing.

```
$ git blame README.md
82496ea3 (kevzettler 2018-02-28 13:37:02 -0800 1) # Gi
82496ea3 (kevzettler 2018-02-28 13:37:02 -0800 2)
89feb84d (Albert So 2018-03-01 00:54:03 +0000 3) This
82496ea3 (kevzettler 2018-02-28 13:37:02 -0800 4)
82496ea3 (kevzettler 2018-02-28 13:37:02 -0800 5) The
82496ea3 (kevzettler 2018-02-28 13:37:02 -0800 6)
89feb84d (Albert So 2018-03-01 00:54:03 +0000 7) Lorem
89feb84d (Albert So 2018-03-01 00:54:03 +0000 8)
eb06faed (Juni Mukherjee 2018-03-01 19:53:23 +0000 9)
eb06faed (Juni Mukherjee 2018-03-01 19:53:23 +0000 10)
548dabed (Juni Mukherjee 2018-03-01 19:55:15 +0000 11)
548dabed (Juni Mukherjee 2018-03-01 19:55:15 +0000 12)
548dabed (Juni Mukherjee 2018-03-01 19:55:15 +0000 13)
```

This is a sample of the first 13 lines of the README.md file. To better understand this output lets break down a line. The following table displays the content of line 3 and the columns of the table indicate the column content.

Id	Author	Timestamp	Line Number	Line Content
89feb84d	Albert So	2018-03-01 00:54:03 +0000	3	This repository is an example of a project with multiple contributors making commits.

If we review the blame output list, we can make some observations. There are three authors listed. In addition to the project's maintainer Kev Zettler, Albert So, and Juni Mukherjee are also listed. Authors are generally the most valuable part of `git blame` output. The timestamp column is also primarily helpful. What the change was is indicated by line content column.

Common Options

```
git blame -L 1,5 README.md
```

The `-L` option will restrict the output to the requested line range. Here we have restricted the output to lines 1 through 5.

```
git blame -e README.md
```

The `-e` option shows the authors email address instead of username.

```
git blame -w README.md
```

The `-w` option ignores whitespace changes. If a previous author has modified the spacing of a file by switching from tabs to spaces or adding new lines this, unfortunately, obscures the output of `git blame` by showing these changes.

```
git blame -M README.md
```

The `-M` option detects moved or copied lines within in the same file. This will report the original author of the lines instead of the last author that moved or copied the lines.

```
git blame -C README.md
```

The `-C` option detects lines that were moved or copied from other files. This will report the original author of the lines instead of the last author that moved or copied the lines.

Git Blame vs Git Log

While `git blame` displays the last author that modified a line, often times you will want to know when a line was originally added. This can be cumbersome to achieve using `git blame`. It requires a combination of the `-w`, `-C`, and `-M` options. It can be far more convenient to use the `git log` command.

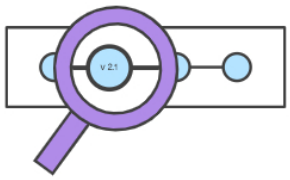
To list all original commits in-which a specific code piece was added or modified execute `git log` with the `-S` option. Append the `-S` option with the code you are looking for. Let's take one of the lines from the README output above to use as an example. Let us take the text "CSS3D and WebGL renderers" from Line 12 of the README output.

```
$ git log -S"CSS3D and WebGL renderers." --pretty=form
e339d3c85 Mario Schuettel Tue Oct 13 16:51:06 2015 +02
509c2cc35 Daniel Tue Sep 8 13:56:14 2015 +0200 Updatec
cb20237cc Mr.doob Mon Dec 31 00:22:36 2012 +0100 Remov
```

This output shows us that content from the README was added or modified 3 times by 3 different authors. It was originally added in commit cb20237cc by Mr.doob. In this example, `git log` has also been prepended with the `--pretty-format` option. This option converts the default output format of `git log` into one that matches the format of `git log`. For more information on usage and configuration options visit the `git log` page.

Summary

The `git blame` command is used to examine the contents of a file line by line and see when each line was last modified and who the author of the modifications was. The output format of `git blame` can be altered with various command line options. Online Git hosting solutions like Bitbucket offer *blame views*, which offer a superior user experience to command line `git blame` usage. `git blame` and git log can be used in combination to help discover the history of a file's contents. The `git log` command has some similar blame functionality, to learn more visit the `git log` overview page.



Next up:

Undoing changes

START NEXT TUTORIAL



Enter Your Email For Git News

ATLASSIAN