

Git Tutorial



Backlog lets you manage projects with Git integration

Learn more →

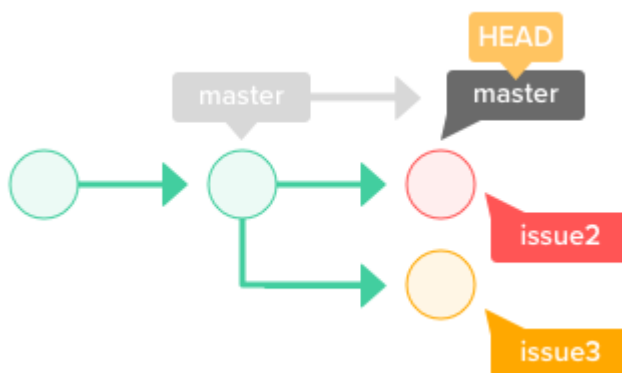
Resolve a merge conflict

Now let's merge branches "issue2" and "issue3" into the master branch.

We will switch over to "master" and merge "issue2" with it.

```
$ git checkout master
Switched to branch 'master'
$ git merge issue2
Updating b2b23c4..8f7aa27
Fast-forward
 myfile.txt | 2 ++
 1 files changed, 2 insertions(+), 0 deletions(-)
```

A fast-forward merge has now been executed.



Let's try to merge "issue3" into "master".

Git Tutorial

```
commit the result.
```

Git has identified a conflict and will not allow you to automatically merge "issue3" with "master". The conflict here pertains to the same line on myfile.txt with different content on both branches. Due to the conflict during the attempt to merge, myfile.txt will look something like this.

```
Git commands even a monkey can understand
add: Register a change in an index
<<<<<<< HEAD
commit: Save the status of an index
=====
pull: Obtain the content of a remote repository
>>>>>>> issue3
```

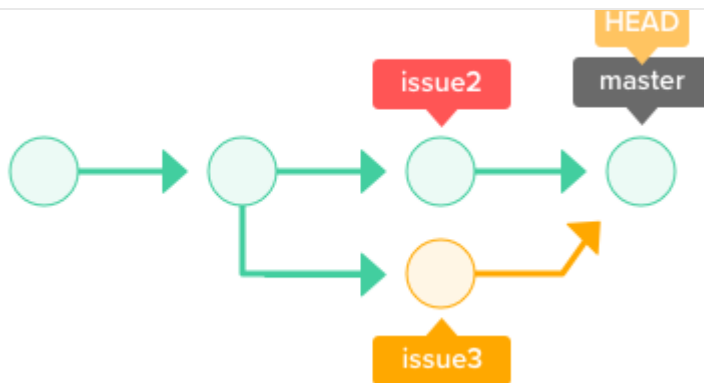
You will find some strange markers being added to myfile.txt. These markers were added by Git, giving us information regarding which line of the file is related to the conflict. We will have to manually fix the conflict as shown below.

```
Git commands even a monkey can understand
add: Register a change in an index
commit: Save the status of an index
pull: Obtain the content of a remote repository
```

Once we are done with resolving the conflict, let's commit the change.

```
$ git add myfile.txt
$ git commit -m "merge issue3 branch"
# On branch master
nothing to commit (working directory clean)
```

Git Tutorial



[Prev](#)

[Next](#)

Backlog has built-in Git repositories for every project

Try it free

Follow us on



[English](#)

[Terms](#)

[Privacy](#)

© 2021 Nulab, Inc. All rights reserved.