Tweet Like 1 G+ Site map

Getting Started | Working with Git | Reference

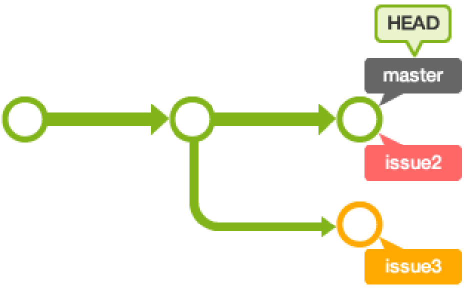
Tutorial 1: Using branch

7. Rebase branch

Another approach we can take to integrate "issue3" branch into the master branch is by using the rebase command. Using rebase, we can streamline and clean our history tree just like how we have described earlier.

Let's start by undoing the previous merge.

```
$ git reset --hard HEAD~
```



Switch over to "issue3" branch and rebase onto the master branch.

```
$ git checkout issue3
Switched to branch 'issue3'
$ git rebase master
First, rewinding head to replay your work on top of it...
Applying: append description of the pull command
Using index info to reconstruct a base tree...
<stdin>:13: new blank line at EOF.
+
warning: 1 line adds whitespace errors.
Falling back to patching base and 3-way merge...
Auto-merging myfile.txt
CONFLICT (content): Merge conflict in myfile.txt
Failed to merge in the changes.
Patch failed at 0001 append description of the pull command

When you have resolved this problem run "git rebase --continue".
If you would prefer to skip this patch, instead run "git rebase --
skip".
To check out the original branch and stop rebasing run "git rebase --
abort".
```

When a conflict occurs during the rebase, you will have to resolve it immediately in order to resume the rebase operation.

```
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
```

Once the conflict is resolved, you can resume rebase with the --continue option. If you wish to quit and rollback the rebase operation, you can do so by passing in the --abort option.

Branch

What's a branch?

How to use a branch

Switching branches

Integrating branches

Branching workflow using topic and integration branch

Tutorial 1: Using branch

0. Preliminary preparation

1. Create branch

2. Switch branches

3. Merge branches

4. Delete branch

5. Work in parallel

6. Resolve merge conflict

7. Rebase branch

Remote repository

Pulling

Fetching

Pushing

Tag

Tag

Tutorial 2: Using tags

0. Preliminary preparation

1. Add a lightweight tag

2. Add an annotated tag

3. Delete a tag

Rewrite a commit

Editing previous commit

Undoing previous commit

Discarding a commit

Applying a commit

Rewriting history

Unify commits from a branch and merge them into a branch

Tutorial 3: Rewriting a commit

1. Commit --amend

2. Revert

3. Reset

4. Cherry-pick

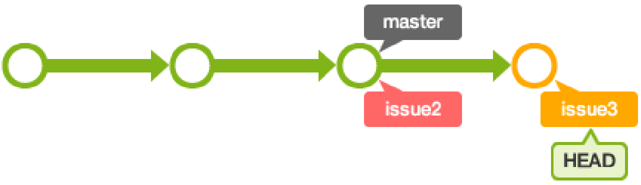
5. Combine commits using rebase -i

6. Amend a commit using rebase -i

Feedback

7. Merge --squash

```
$ git add myfile.txt
$ git rebase --continue
Applying: append description of the pull command
```

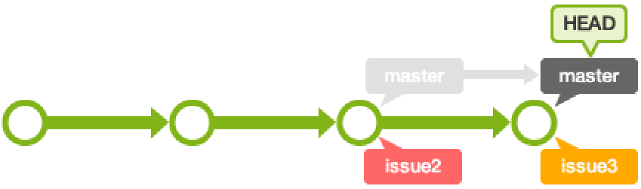


With the "issue3" branch rebased onto "master", we can now issue a fast-forward merge.

Switch over to the master branch and merge "issue3" with "master".

```
$ git checkout master
Switched to branch 'master'
$ git merge issue3
Updating 8f7aa27..96a0ff0
Fast-forward
 myfile.txt |    1 +
 1 files changed, 1 insertions(+), 0 deletions(-)
```

The content of myfile.txt should now be identical to the one that we got from the previous merge. The revision history now should look like the following.



Previous Next

Powered by

English

Feedback