EcomLabs

STEP-BY-STEP GUIDE

A git workflow for AI coding agents

The git workflow for asking an AI agent for changes without fear: a safe point, the agent's branch, review, small commits and how to go back. We ran every command in a throwaway repository, and the cheat sheet is at the end.

FOR	Claude Code, Cursor or any agent that edits your repository
TESTED WITH	git 2.54 · Node 22 · Windows 11 (Git Bash and PowerShell)
UPDATED	October 30, 2026
DOCS	git-scm.com/docs
ARTICLE	ecomlabs.dev/g/git-en
AUTHOR	Luis Morales · @luismoralesdev

Contents

Step 01	Leave a safe point
Step 02	Create the agent's branch
Step 03	Ask for the change
Step 04	Review before you commit
Step 05	Commit in small pieces
Step 06	If it went wrong, go back
Step 07	Merge or drop the branch
Cheat sheet	Which command for each case
When you are done	Checklist

STEP 01

Leave a safe point

The output in this guide comes from a throwaway repository on git 2.54: a store where the agent adds a discount to the cart. To practise without touching your project, create one with `git init -b main`. Before you ask for anything, check for uncommitted work:

```
$ git status --short
 M README.md
```

One line per file: `M` is modified and not staged, `??` is a new file git does not track, and no output means nothing is pending. Commit your work:

```
$ git add -A
$ git commit -m "Document how to run the tests"
[main 59ba9a5] Document how to run the tests
 1 file changed, 2 insertions(+)
```

Note the hash (`59ba9a5`): it is the point you can go back to.

STEP 02

Create the agent's branch

```
$ git switch -c agent/bulk-discount
Switched to a new branch 'agent/bulk-discount'
```

`-c` creates the branch and switches to it in one step. The `agent/` prefix is a convention: it tells you which branches you can throw away without a second thought.

STEP 03

Ask for the change

Tell the agent what you expect at the end:

```
Add a 10% discount from 3 items in cartTotal, with tests.
Do not commit or push: when you are done, list the files you touched and run npm test.
```

If it commits anyway, it is on its own branch: step 06 covers how to undo that.

STEP 04

Review before you commit

```
$ git status --short
 M src/cart.js
?? test/discount.test.js
$ git diff --stat
 src/cart.js | 5 ++++-
 1 file changed, 4 insertions(+), 1 deletion(-)
```

`git diff` leaves out the new test: it compares your files with the index, and a file git does not track is not in the index. Read the full diff with `git diff`, then, to see new files too:

```
$ git add -A
$ git diff --staged --stat
 src/cart.js           |  5 ++++-
 test/discount.test.js | 15 +++++++++++++++
 2 files changed, 19 insertions(+), 1 deletion(-)
```

Run the tests (`npm test`; 4 of 4 in the demo). In the diff, look for files you did not expect, leftover debugging (a `console.log` in the demo) and existing tests that changed so they would pass.

STEP 05

Commit in small pieces

Leave out what you did not ask for. Unstage the file and pick hunk by hunk:

```
$ git restore --staged src/cart.js
$ git add -p src/cart.js
(1/2) Stage this hunk [y,n,q,a,d,k,K,j,J,g,/,e,p,P,?]?
(2/2) Stage this hunk [y,n,q,a,d,K,J,g,/,s,e,p,P,?]?
$ git commit -m "Add 10% discount from 3 items"
[agent/bulk-discount 01999c6] Add 10% discount from 3 items
 2 files changed, 18 insertions(+), 1 deletion(-)
 create mode 100644 test/discount.test.js
$ git restore src/cart.js
```

In `git add -p`: `y` stages the hunk, `n` leaves it out, `q` quits and `s` splits a hunk when it can. In the demo we answered `n` to the `console.log` and `y` to the discount. The last `git restore` drops what was left out of the commit.

Messages: a first line of no more than 50 characters that says what changes. `git log --oneline` then reads as your list of safe points.

STEP 06

If it went wrong, go back

SITUATION	COMMAND
One file came out wrong	`git restore <file>`
You staged something by mistake	`git restore --staged <file>`
Everything uncommitted has to go	`git reset --hard`
New files or folders (reset leaves them)	`git clean -nd`, then `git clean -fd`
The agent's commit has to go	`git reset --hard <safe-point hash>`

> **CAUTION**
> `git reset --hard` deletes every uncommitted change to tracked files, and work that was never committed cannot be recovered, not even with `git reflog`. Use it only on the agent's working branch, never on `main` or a shared branch, and always after a look at `git status`.

In the demo, `src/tmp-debug.js` survived `git reset --hard` until `git clean -f`; run `-n` first to see what would go. If you regret a reset to an earlier commit, the commits are still in the reflog (90 days by default; 30 once no branch points to them):

```
$ git reset --hard 59ba9a5
HEAD is now at 59ba9a5 Document how to run the tests
$ git reflog -3
59ba9a5 HEAD@{0}: reset: moving to 59ba9a5
01999c6 HEAD@{1}: reset: moving to HEAD
01999c6 HEAD@{2}: commit: Add 10% discount from 3 items
$ git reset --hard 'HEAD@{1}'
HEAD is now at 01999c6 Add 10% discount from 3 items
```

The quotes around `'HEAD@{1}'` are for PowerShell, where the command fails without them; in bash and zsh they make no difference.

STEP 07

Merge or drop the branch

```
$ git switch main
$ git merge --ff-only agent/bulk-discount
Updating 59ba9a5..01999c6
Fast-forward
$ git branch -d agent/bulk-discount
Deleted branch agent/bulk-discount (was 01999c6).
```

`--ff-only` only merges if `main` has not moved on; if it has, it refuses and you decide calmly how to merge. To drop an attempt that is no good, `git branch -D agent/<task>`: `-d` refuses branches that are not merged.

If the change is already on `main` or pushed, do not reset: `git revert --no-edit <commit>` undoes it with a new commit.

CHEAT SHEET

Which command for each case

YOU WANT TO…	COMMAND
See what changed	`git status --short` · `git diff`
Leave a safe point	`git add -A` · `git commit -m "..."`
Branch for the agent	`git switch -c agent/<task>`
Commit only part of a file	`git add -p <file>`
Discard one file	`git restore <file>`
Unstage a file	`git restore --staged <file>`
Discard all uncommitted work (careful)	`git reset --hard`
Delete new files	`git clean -n` · `git clean -f` (with folders: `-nd`, `-fd`)
Go back to a commit (careful)	`git reset --hard <commit>`
Recover after a reset	`git reflog` · `git reset --hard 'HEAD@{1}'`
Merge the branch	`git switch main` · `git merge --ff-only agent/<task>`
Delete the branch	`git branch -d` (merged) · `git branch -D` (dropped)
Undo something shared	`git revert --no-edit <commit>`

WHEN YOU ARE DONE

Checklist

- [ ] A clean `git status`, or your work committed, before asking; the agent on its own branch.
- [ ] You checked `git status --short`, the `??` lines and the full diff.
- [ ] The tests pass and no existing test changed without a reason.
- [ ] `git reset --hard` and `git clean -f` only on the agent's branch.
- [ ] You merged with `--ff-only` and deleted the branch.