

# Squasher un historique

La méthode la plus courante pour écraser les commits consiste à utiliser un rebase interactif. Nous le démarrons à l'aide de la commande :

```
git rebase -i HEAD~<number_of_commits>
```

Remplacez par le nombre de commits que nous voulons écraser.

L'exécution de cette commande ouvrira un éditeur de ligne de commande interactif.

```
pick 6a8038b Create initial form
pick 78fdad0 Implement form validation
pick 898a3c3 Implement form styling
pick 150d7b5 Fix validation bug

# Rebase 2831881..150d7b5 onto 2831881 (4 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
#                               commit's log message, unless -C is used, in which case
#                               keep only this commit's message; -c is same as -C but
#                               opens the editor
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
#       create a merge commit using the original merge commit's
#       message (or the oneline, if no original merge commit was
```

La partie supérieure affiche les commits, tandis que la partie inférieure contient des commentaires sur la façon d'écraser les commits.

Nous avons quatre commits. Pour chacun d'entre eux, nous devons décider de la commande à exécuter. Nous nous intéressons aux commandes pick (p) et squash (s). Pour écraser ces quatre commits en un seul, nous pouvons choisir le premier et écraser les trois autres.

Nous appliquons les commandes en modifiant le texte précédant chaque livraison, en particulier en remplaçant pick par s ou squash pour les deuxième, troisième et quatrième livraisons.

```

pick 6a8038b Create initial form
s 78fdad0 Implement form validation
s 898a3c3 Implement form styling
s 150d7b5 Fix validation bug

# Rebase 2831881..150d7b5 onto 2831881 (4 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
#                               commit's log message, unless -C is used, in which case
#                               keep only this commit's message; -c is same as -C but
#                               opens the editor
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
#     create a merge commit using the original merge commit's
#     message (or the oneline, if no original merge commit was
-- INSERT --

```

Une fois terminé, sauvegarder puis fermer l'éditeur.

---

Revision #1

Created 9 June 2025 16:06:40 by Arsène Lapostolet

Updated 9 June 2025 16:08:47 by Arsène Lapostolet