

Git Basic Commands

Cheat Sheet

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Revision**  **History** | **Revision Description** | **Author** | **Comments** |
| March 14, 2019 | 1.0.0 | First draft |  | Initial Draft |

Table of Contents

[1 Overview 4](#_Toc3472206)

[1.1 Setup 4](#_Toc3472207)

[1.2 Help 4](#_Toc3472208)

[1.3 General 4](#_Toc3472209)

[1.4 Reset 5](#_Toc3472210)

[1.5 Update and Delete 5](#_Toc3472211)

[1.6 Branch 6](#_Toc3472212)

[1.7 Merge 6](#_Toc3472213)

[1.8 Stash 7](#_Toc3472214)

[1.9 Gitignore and Gitkeep 7](#_Toc3472215)

[1.10 Log 7](#_Toc3472216)

[1.11 Compare 8](#_Toc3472217)

[1.12 Releases and Version Tags 9](#_Toc3472218)

[1.13 Collaborate 9](#_Toc3472219)

[1.14 Archive 10](#_Toc3472220)

# Overview

This document provides Git Bitbucket basic commands with examples which are useful for the developer’s community.

The Git Bitbucket commands are presented in the following sections, navigate to the section(s) you require and familiarize the commands with examples:

* Setup
* Help
* General
* Reset
* Update and Delete
* Branch
* Merge
* Stash
* Gitignore and Gitkeep
* Log
* Compare
* Releases and Version Tags
* Collaborate
* Archive

## Setup

|  |  |
| --- | --- |
| **Git Command** | **Description** |
| which git | See where Git is located |
| git –version | Get the version of Git |
| git config --global alias.st status | Create an alias (shortcut) for git status |

## Help

|  |  |
| --- | --- |
| **Git Command** | **Description** |
| git help | Help |

## General

|  |  |
| --- | --- |
| **Git Command** | **Description** |
| git init | Initialize Git |
| git add . | Get everything ready to commit |
| git add index.html | Get custom file ready to commit |
| -m "Message" | Commit changes: git commit |
| git commit -m "Title" -m "Description..." | Commit changes with title and description |
| -am "Message" | Add and commit in one step: git commit |
| git rm index.html | Remove files from Git |
| git add -u | Update all changes |
| git rm --cached index.html | Remove file but do not track anymore |
| git mv index.html dir/index\_new.html | Move or rename files |
| git checkout -- index.html | Undo modifications (restore files from latest committed version) |
| git checkout 6eb715d -- index.html | Restore file from a custom commit (in current branch) |

## Reset

|  |  |
| --- | --- |
| **Git Command** | **Description** |
| git revert 073791e7dd71b90daa853b2c5acc2c925f02dbc6 | Go back to commit |
| git reset --soft 073791e7dd71b90daa853b2c5acc2c925f02dbc6 | Soft reset (move HEAD only; neither staging nor working dir is changed) |
| git reset --soft HEAD~ | Undo latest commit |
| git reset --mixed 073791e7dd71b90daa853b2c5acc2c925f02dbc6 | Mixed reset (move HEAD and change staging to match repo - does not affect working dir) |
| git reset --hard 073791e7dd71b90daa853b2c5acc2c925f02dbc6 | Hard reset (move HEAD and change staging dir and working dir to match repo) |
| git checkout @ -- index.html | Hard reset of a single file (@ is short for HEAD) |

## Update and Delete

|  |  |
| --- | --- |
| **Git Command** | **Description** |
| git clean -n | Test-Delete untracked files |
| git clean -f | Delete untracked files (not staging) |
| git reset HEAD index.html | Unstage (undo adds) |
| git commit --amend -m "New Message" | Update most recent commit (also update the commit message) |

## Branch

|  |  |
| --- | --- |
| **Git Command** | **Description** |
| git branch | Show branches |
| git branch branchname | Create branch |
| git checkout branchname | Change to branch |
| git checkout -b branchname | Create and change to new branch |
| git branch -m branchname new\_branchname  -or-  --move branchname new\_branchname | Rename branch  -or-  git branch |
| git branch –merged | Show all completely merged branches with current branch |
| git branch -d branchname  -or-  git branch --delete branchname  git branch -D branch\_to\_delete | Delete merged branch (only possible if not HEAD)  -or-  Delete not merged branch |

## Merge

|  |  |
| --- | --- |
| **Git Command** | **Description** |
| git merge branchname | True merge (fast forward) |
| git merge --ff-only branchname | Merge to master (only if fast forward) |
| git merge --no-ff branchname | Merge to master (force a new commit) |
| git merge –abort | Stop merge (in case of conflicts) |
| git reset --merge // prior to v1.7.4 | Stop merge (in case of conflicts) |
| git reset --hard origin/master | Undo local merge that hasn't been pushed yet |
| git cherry-pick 073791e7 | Merge only one specific commit |
| git checkout branchname » git rebase master  -or-  git merge master branchname | Rebase (The rebase moves all the commits in master onto the tip of branch name) |
| git rebase –abort | Cancel rebase |
| git rebase -i HEAD~3 | Squash multiple commits into one |
| git merge --squash branchname  (commit afterwards) | Squash-merge a feature branch (as one commit) |

## Stash

|  |  |
| --- | --- |
| **Git Command** | **Description** |
| git stash save "Message" | Put in stash |
| git stash list | Show stash |
| git stash show stash@{0} | Show stash stats |
| git stash show -p stash@{ | Show stash changes:  0} |
| git stash pop stash@{0} | Use custom stash item and drop it |
| git stash apply stash@{0} | Use custom stash item and do not drop it |
| git stash apply --index | Use custom stash item and index |
| git stash branch new\_branch | Create branch from stash |
| git stash drop stash@{0} | Delete custom stash item |
| git stash clear | Delete complete stash |

## Gitignore and Gitkeep

|  |  |
| --- | --- |
| **Git Command** | **Description** |
| nano .gitignore | Add or edit gitignore |
| touch dir/.gitkeep | Track empty dir |

## Log

|  |  |
| --- | --- |
| **Git Command** | **Description** |
| git log | Show commits |
| git log –oneline | Show one line-summary of commits |
| git log --format=oneline | Show one line-summary of commits with full SHA-1 |
| git log --oneline -3 | Show one line-summary of the last three commits |
| git log --author="Sven" git log --grep="Message" git log --until=2013-01-01 git log --since=2013-01-01 | Show only custom commits |
| git log --format=short git log --format=full git log --format=fuller git log --format=email git log --format=raw | Show only custom data of commit |
| git log -p | Show changes |
| git log 6eb715d.. index.html | Show every commit since special commit for custom file only |
| git log -p 6eb715d.. index.html | Show changes of every commit since special commit for custom file only |
| git log --stat --summary | Show stats and summary of commits |
| git log –graph | Show history of commits as graph |
| git log --oneline --graph --all --decorate | Show history of commits as graph-summary |

## Compare

|  |  |
| --- | --- |
| **Git Command** | **Description** |
| git diff | Compare modified files |
| git diff --color-words index.html | Compare modified files and highlight changes only |
| git diff –staged | Compare modified files within the staging area |
| git diff master..branchname | Compare branches |
| git diff --color-words master..branchname^ | Compare branches like above |
| git diff 6eb715d git diff 6eb715d..HEAD git diff 6eb715d..537a09f | Compare commits |
| git diff 6eb715d index.html git diff 6eb715d..537a09f index.html | Compare commits of file |
| git diff -b 6eb715d..HEAD or: git diff --ignore-space-change 6eb715d..HEAD | Compare without caring about spaces |
| git diff -w 6eb715d..HEAD or: git diff --ignore-all-space 6eb715d..HEAD | Compare without caring about all spaces |
| git diff --stat --summary 6eb715d..HEAD | Useful comparings |
| git blame -L10,+1 index.html | Blame |

## Releases and Version Tags

|  |  |
| --- | --- |
| **Git Command** | **Description** |
| git tag | Show all released versions |
| git tag -l -n1 | Show all released versions with comments |
| git tag v1.0.0 | Create release version |
| git tag -a v1.0.0 -m 'Message' | Create release version with comment |
| git checkout v1.0.0 | Checkout a specific release version |

## Collaborate

|  |  |
| --- | --- |
| **Git Command** | **Description** |
| git remote | Show remote |
| git remote -v | Show remote details |
| git remote add upstream https://github.com/user/project.git | Add remote upstream from GitHub project |
| git remote add upstream ssh://root@123.123.123.123/path/to/repository/.git | Add remote upstream from existing empty project on server |
| git fetch upstream | Fetch |
| git fetch upstream branchname:local\_branchname | Fetch a custom branch |
| git merge upstream/master | Merge fetched commits |
| git remote rm origin | Remove origin |
| git branch -r | Show remote branches |
| git branch -a | Show all branches (remote and local) |
| git checkout -b local\_branchname upstream/remote\_branchname | Create and checkout branch from a remote branch |
| git diff origin/master..master | Compare |
| git push -u origin master | Push (set default with -u) |
| git push origin master | Push |
| `git push origin master --force | Force-Push |
| git pull | Pull |
| git pull origin branchname | Pull specific branch |
| git fetch upstream pull/ID/head:new-pr-branch | Fetch a pull request on GitHub by its ID and create a new branch |
| git clone https://github.com/user/project.git or: git clone ssh://user@domain.com/~/dir/.git | Clone to localhost |
| git clone https://github.com/user/project.git ~/dir/folder | Clone to localhost folder |
| git clone -b branchname https://github.com/user/project.git | Clone specific branch to localhost |
| git push origin :branchname or: git push origin --delete branchname | Delete remote branch (push nothing) |

## Archive

|  |  |
| --- | --- |
| **Git Command** | **Description** |
| git archive --format zip --output filename.zip master | Create a zip-archive |
| git log --author=apandey --all > log.txt | Export/write custom log to a file |