



GitHub

Tim Choh

How to Setup Github

- [GitHub](#)
- make an account
- check email and confirm account
- log in online to make sure everything is set up

Git GUI (For those not wanting to work with Bash)

- two main types for OSX and Windows:
 - [Github GUI](#) (for respective OS)
 - SourceTree
- For Linux:
 - Gigggle
 - gitg

Git Terminal

When downloading the GUI of your choice, make sure to download the included terminal commands as well (important if you want to learn how to use the terminal commands)

Git Terminal

If you are on a Linux-based machine:

`sudo apt-get install git` → Debian based (includes Ubuntu)

`sudo yum install git` → Fedora

Windows also supports Bash through the anniversary edition

[Enabling Bash tutorial](#)

Personal Note

I have used the terminal version ever since I started and I have never switched to the GUI because I got very used to terminal.

That being said, I can help as much as I can with the GUI version, but my “expertise” is in the terminal.

Stages in Git

1. Changed files
2. Staging phase
3. Commit phase
4. Push phase

How To Git

Once Git is installed on machine and account is made:

Go to project you want and clone (creates a local version of the repo on Github) or fork (creates a personal version that's independent of the one on GitHub)

How To Git Cont.

The image shows a GitHub repository page for 'timchoh585 / PotKiller-android'. The repository has 4 watches, 0 stars, and 0 forks. It is a Hack Illinois android app using the MIT license, with 1 contributor. The page shows the commit history with the latest commit 'Update README.md' by timchoh585 4 minutes ago. A dropdown menu for 'Clone or download' is open, showing options to 'Clone with HTTPS' (using the URL https://github.com/timchoh585/PotKiller-an) or 'Use SSH'. There are also buttons for 'Open in Desktop' and 'Download ZIP'.

timchoh585 / PotKiller-android

Unwatch 4 Star 0 Fork 0

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

Hack Illinois android app — Edit

2 commits 6 branches 0 releases 1 contributor MIT

Branch: master New pull request

Create new file Upload files Find file Clone or download

timchoh585 committed on GitHub Update README.md Latest commit 3ccd976 4 minutes ago

LICENSE	Initial commit	7 months ago
README.md	Update README.md	4 minutes ago

README.md

PotKiller-android

Hack Illinois android app A way to REPORT Potholes around the city and post them to the City Of Chicago potholes through their API

1 contributor MIT

Create new file Upload files Find file Clone or download

Clone with HTTPS ? Use SSH

Use Git or checkout with SVN using the web URL.

https://github.com/timchoh585/PotKiller-an

Open in Desktop Download ZIP

How To Git Cont.

Take the URL and paste it into terminal

```
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop$ git clone https://github.com/timchoh585/PotKiller-android.git
Cloning into 'PotKiller-android'...
remote: Counting objects: 331, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 331 (delta 0), reused 0 (delta 0), pack-reused 328
Receiving objects: 100% (331/331), 835.87 KiB | 934.00 KiB/s, done.
Resolving deltas: 100% (113/113), done.
Checking connectivity... done.
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop$
```

Double check URL if this is not what you get

How To Git Cont.

Double check that all the files are there

```
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ ls  
LICENSE  README.md  
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$
```

If files are missing, reclone the repo OR “git pull” to attempt to get your local repo with what GitHub has

How To Git Cont.

Stages of Git:

- 1) Newly added files (or changed files) will show up in “RED” when “git status” is run

```
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ ls
hello.txt  LICENSE  README.md
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        hello.txt

nothing added to commit but untracked files present (use "git add" to track)
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$
```

How To Git Cont.

Stages of Git:

2) staging phase

When you “add” the files, the files will show up in “GREEN” when “git status” is run

2 ways to add a file:

- 1) git add <filename> → more precise
- 2) git add -a → add all files in current directory

```
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ git add hello.txt
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   hello.txt

tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$
```

```
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ git add -A
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   hello.txt
```

How To Git Cont.

Stages of Git:

3) commit files

2 ways to do this:

- 1) Git commit → will take you to nano and will ask you to put a message to describe the need for a commit
- 2) Git commit -m "<message>" → add message in-line

```
GNU nano 2.2.6      File: /mnt/c/Users/timch/Desktop/PotKiller-android/.git/COMMIT_EDITMSG

some message as to why you want to commit the files

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Your branch is ahead of 'origin/master' by 1 commit.
#   (use "git push" to publish your local commits)
#
# Changes to be committed:
#   modified:   hello.txt
#
```

```
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ git add hello.txt
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   hello.txt

tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$
```

How To Git Cont.

Stages of Git:

4) push files

Push files to
branch (default
is master)

(first time will
require user
auth)

```
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

    git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

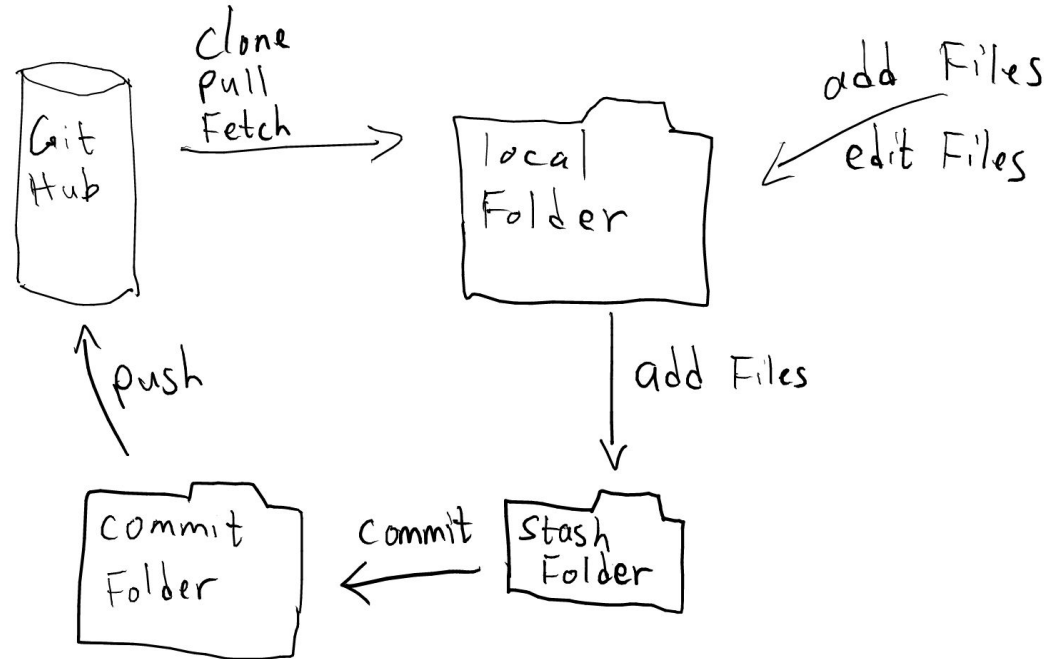
In Git 2.0, Git will default to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Username for 'https://github.com': timchoh585
Password for 'https://timchoh585@github.com':
Counting objects: 12, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 553 bytes | 0 bytes/s, done.
Total 6 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/timchoh585/PotKiller-android.git
   3ccd976..3579adf master -> master
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$
```

Crude drawing of How it all works

The “stash folder” and “commit folder” are hidden files that are members of the “.git” hidden folder and part of git’s %appdata% file



Branches!

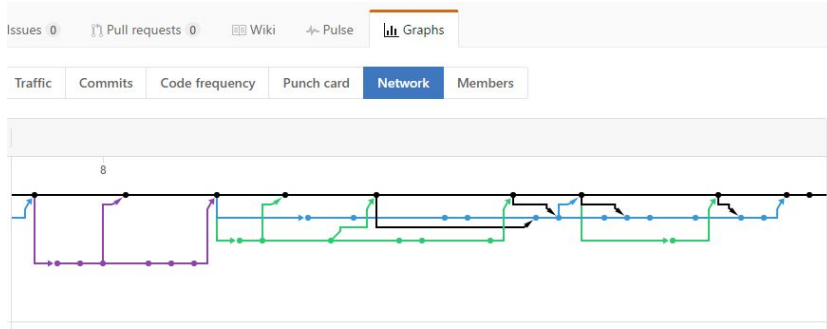
Git has this thing called “branch”.

Master branch is the head branch (release branch)

Every other branch is meant to fix (or add) something

When a branch is done, a “pull request” is made. If that gets accepted, then that branch gets merged with another branch

GitHub has a visual to be able to see the branches. (Black is always the master branch)



Branches cont.

You can also switch branches to see the work on other branches

Hack Illinois android app — Edit

4 commits

6 branches

0 releases

Branch: master ▼

New pull request

Switch branches/tags

Find or create a branch...

Branches Tags

dev

layout

map_button

maps

✓ master

merged

Branches cont.

You can make a new branch with: `git checkout -b <branch name>`

```
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ git checkout -b test
Switched to a new branch 'test'
```

`-b` is to make a new branch

Without it, you can switch between branches (`git checkout <branch name>`)

```
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.
```

It will also tell you if the branch is up-to-date with what GitHub has stored

Branches cont.

If you want to see what branches you have:

```
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ git branch -a
* master
test
remotes/origin/HEAD -> origin/master
remotes/origin/dev
remotes/origin/layout
remotes/origin/map_button
remotes/origin/maps
remotes/origin/master
remotes/origin/merged
```

Key:

Green - current branch

White - locally saved branches

Red - branches only stored on GitHub (will have to pull the branch to get it local)

Branches cont.

If you want to get a branch from origin (GitHub version)

Make a local branch (make sure it's the same name as the one on origin)

Pull branch into the local branch

```
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ git branch
* master
test
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ git checkout -b dev
Switched to a new branch 'dev'
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ git pull origin dev
From https://github.com/timchoh585/PotKiller-android
* branch          dev          -> FETCH_HEAD
Already up-to-date.
tim@TIMC-V15NITRO:/mnt/c/Users/timch/Desktop/PotKiller-android$ git branch -a
* dev
master
test
remotes/origin/HEAD -> origin/master
remotes/origin/dev
remotes/origin/layout
remotes/origin/map_button
remotes/origin/maps
remotes/origin/master
remotes/origin/merged
```

Git Commands

- get help
 - git help
- tell Git who you are - configure name and email
 - git config --global user.name "timchoh585"
 - git config --global user.email "timchoh@gmail.com"
- create new local repo
 - git init

Git Commands cont.

- check out repo - create copy of repo
 - `git clone /path/to/repo`
- add files - add one or more files to staging
 - `git add <filename>`
 - `git add -A`
 - `git add *`
- push - send changes to remote branch
 - `git push _____`

Git Commands cont.

- status - get status of git repo of branch
 - git status
- branches - making “second repo” in main repo
 - git checkout -b nameofbranch - make new branch
 - git checkout nameofbranch - switch branches
 - git branch - list all branches
 - git push origin branchname - local to remote

Git Commands cont.

- fetch
 - `git fetch origin` - remove local changes and commits and fetch origin branch
 - `git reset --hard <branch name>` - remove old changes and revert to old push
- pull
 - `git pull` - fetch and merge changes to local from remote

Git Commands cont.

<https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html>

<https://training.github.com/kit/downloads/github-git-cheat-sheet.pdf>