

Git Repository

簡介

Git 儲存庫 (Git Repository) 大致可以分為三種

- 本地的儲存庫 (local repository)
- 共用的儲存庫 (shared repository)
- 遠端的儲存庫 (remote repository)

其實都是差不多的東西，Git 是分散式的版本控制，所以只是放置的位置、使用者不同而已。

1. Local repository

```
$ git init myproject
$ cd myproject
$ ls
.git/
```

用 `init` 不帶參數新增的專案包含：工作目錄 (Working Directory) + Git 儲存庫 (Git Repository)

- 工作目錄：放置程式的地方，ex: `.py`、`.js` 的檔案
- Git 儲存庫：`.git` 資料夾

2. Remote repository

常見的託管平台：`github`、`gitlab`、`BitBucket`、...etc

遠端已經存在的專案

```
$ git clone http://github.com/hbdoy/xxx
```

將 local 的 repo 首次提交

```
$ git remote add origin https://github.com/hbdoy/xxx.git
$ git push -u origin master

// also you can use
$ git push --all
$ git push --tags
or
$ git push --mirror
```

若遠端的 repo 網址改變，可以修改 `origin` 位址

```
$ git remote set-url origin https://github.com/hbdoy/new_xxx.git
```

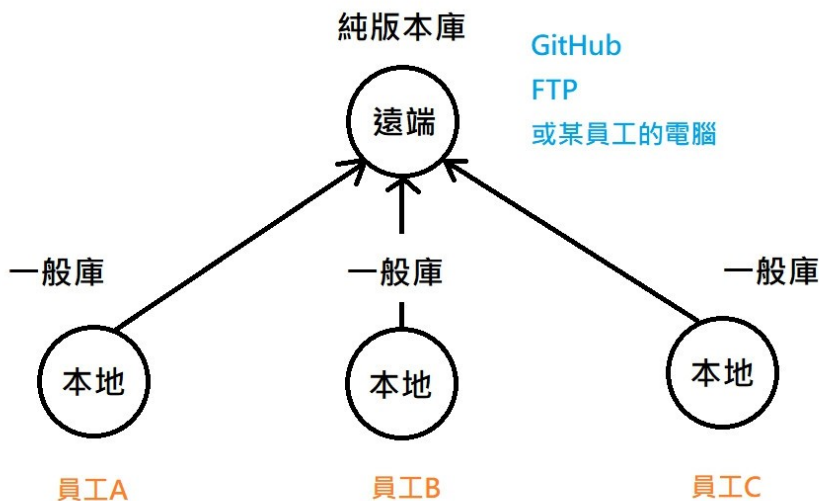
3. Shared repository

和 remote repository 差不多，都可供多人同時使用開發，實作上也可以理解為 **Git Bare Repo (裸庫)**

```
$ git init --bare bareproject
```

不同於上面，Bare Repo 只有 Git 儲存庫 (Git Repository) 而已，也就是**沒有工作目錄**，因為沒有程式檔案，所以你無法在上面 `git add`、`git commit`，他的用意也不是要讓你在上面開發的。

以下是一個概念，沒有限定要用幾台機器



以上圖為例

員工 A、B、C 的電腦裡有各自的 Git 版本記錄了今天完成的項目

該如何合併三人的程式呢？

大家下班前會將自己的部分 `push` 到遠端，隔天上班在 `pull` 下來，這樣大家的程式就合併完成了，這就是 Git 的操作概念

遠端究竟要用純版本庫還是一般版本庫？

一般建議使用純版本庫。如果使用一般版本庫，會因為遠端 `checkout` 到某個分支，Git 為了避免覆蓋到遠端的工作目錄，而**拒絕本地推送同名的分支**。

兩者差在工作目錄而已，把他們分開，是為了避免 `checkout` 衝突的問題，這也方便解釋遠端和本地 的概念

Bare Repo 雖說有 Git 儲存庫，但他沒有 `.git` 資料夾，而是將裡面的內容直接放到專案根目錄下，專案的資料夾名稱習慣會在最後面加上 `'.git'`

EX: bareproject.git

bareproject.git/

名稱	修改日期	類型	大小
hooks	2020/4/7 下午 02:35	檔案資料夾	
info	2020/4/7 下午 02:35	檔案資料夾	
objects	2020/4/7 下午 02:35	檔案資料夾	
refs	2020/4/7 下午 02:35	檔案資料夾	
config	2020/4/7 下午 02:35	檔案	1 KB
description	2020/4/7 下午 02:35	檔案	1 KB
HEAD	2020/4/7 下午 02:35	檔案	1 KB

對比一般含有工作目錄的 Repo

名稱	修改日期	類型	大小
.git	2020/4/9 下午 03:47	檔案資料夾	
.nuget	2020/4/7 上午 11:59	檔案資料夾	
.vs	2020/4/9 下午 02:04	檔案資料夾	
data	2020/4/7 上午 11:59	檔案資料夾	
GSS.BB13.API	2020/4/8 上午 11:53	檔案資料夾	
GSS.BB13.API.Client	2020/4/8 上午 11:53	檔案資料夾	
GSS.BB13.API.Client.Test	2020/4/7 下午 12:11	檔案資料夾	
GSS.BB13.AR	2020/4/8 上午 11:53	檔案資料夾	
GSS.BB13.Common	2020/4/9 下午 03:47	檔案資料夾	
GSS.BB13.FA	2020/4/7 上午 11:59	檔案資料夾	
GSS.BB13.GL	2020/4/9 下午 03:47	檔案資料夾	
GSS.BB13.Model	2020/4/9 下午 03:47	檔案資料夾	
GSS.BB13.Provision	2020/4/8 上午 11:53	檔案資料夾	
GSS.BB13.Resources	2020/4/8 上午 11:53	檔案資料夾	
GSS.BB13.Security	2020/4/9 下午 03:47	檔案資料夾	
GSS.BB13.WindowsAzure	2020/4/7 上午 11:59	檔案資料夾	
Libs	2020/4/9 上午 11:48	檔案資料夾	
packages	2020/4/9 上午 11:47	檔案資料夾	
SQLScript	2020/4/8 上午 01:01	檔案資料夾	
.gitattributes	2020/4/8 上午 01:01	文字文件	146 KB
.gitignore	2020/4/9 下午 03:47	文字文件	18 KB
build.xml	2020/4/7 上午 11:59	XML Document	2 KB
GSS.BB13.API.Client.sln	2020/4/7 上午 11:59	Visual Studio Sol...	2 KB
GSS.BB13.API.Client.Test.sln	2020/4/7 上午 11:59	Visual Studio Sol...	2 KB

Bare Repo 的儲存庫內容物其實不完全等於 **.git** 中的東西，**.git** 還會包含一些工作目錄下的資訊，而這些資訊不會放到 **Bare Repo** 中

用途

Bare Repo 像是一個 **Git Server**，可以讓別人 **clone** 下來開發，開發完再 **push** 回去，比起上面兩者是用來開發(工作)用的，**Bare Repo** 是用來分享(共享)的，而且沒有存著程式檔案，整個專案資料夾容量較小。

情境

在 Linux 上建立 Bare Repo，讓數個開發者透過 ssh 來存取這個資料夾，進行 clone & push 等操作

如何在 Linux 上建立 Remote Bare Git Repo 可參考：

- <https://blog.gtwang.org/linux/linux-git-server-using-ssh/>
- <https://stevenitlife.blogspot.com/2015/02/repository.html>
 1. 將目錄的權限改為 **770**，讓同群組的成員也可以存取
 2. 將開發人員帳號加入 git 群組

```
$ usermod -a -G <group-name> <user-name>
# '-G' 參數時，usermod 會將帳號退出原來加入的群組
# '-a' 參數，會保留原來的群組設定
```

或是建立在其他資料夾同步軟體，EX: Google Sync、Dropbox

差異

- **一般 Git Repo** 用來開發，工作目錄存著程式檔案、儲存庫記錄著所有檔案的版本訊息
- **Bare Repo** 用來單純紀錄檔案的版本訊息，中心點上建立一個 Bare Repo，所有專案成員可以 push 自己編輯的版本訊息到這個 Repo

Bare Repo 常用指令

1. 建立

Local

```
$ git --bare init xxx
```

將 Remote Repo 轉成 Bare Repo

```
$ git clone --bare http://github.com/hbdoy/myproject
```

```
D:\GSS_project>git clone --bare https://github.com/hbdoy/roulette
Cloning into bare repository 'roulette.git'...
remote: Enumerating objects: 194, done.
remote: Counting objects: 100% (194/194), done.
remote: Compressing objects: 100% (145/145), done.
remote: Total 194 (delta 103), reused 127 (delta 45), pack-reused 0iving objects: 52% (101/194), 10.1
iB/
Receiving objects: 100% (194/194), 10.36 MiB | 2.85 MiB/s, done.
Resolving deltas: 100% (103/103), done.
```

(D:) > GSS_project > roulette.git

名稱	修改日期	類型	大小
hooks	2020/4/7 下午 01:49	檔案資料夾	
info	2020/4/7 下午 01:49	檔案資料夾	
objects	2020/4/7 下午 01:49	檔案資料夾	
refs	2020/4/7 下午 01:49	檔案資料夾	
config	2020/4/7 下午 01:56	檔案	1 KB
description	2020/4/7 下午 01:49	檔案	1 KB
HEAD	2020/4/7 下午 01:49	檔案	1 KB
packed-refs	2020/4/7 下午 01:49	檔案	1 KB

2. Clone

Clone From Bare Repo

此處示範 clone 本地上的 bare repo

```
$ git clone roulette.git
```

```
D:\GSS_project>git clone roulette.git
Cloning into 'roulette'...
done.
```

可以看到 Bare Repo Clone 出來的專案自動長出工作目錄

(D:) > GSS_project > roulette >

名稱	修改日期	類型	大小
.git	2020/4/7 下午 03:15	檔案資料夾	
css	2020/4/7 下午 03:15	檔案資料夾	
img	2020/4/7 下午 03:15	檔案資料夾	
js	2020/4/7 下午 03:15	檔案資料夾	
media	2020/4/7 下午 03:15	檔案資料夾	
.gitignore	2020/4/7 下午 03:15	文字文件	0 KB
about.html	2020/4/7 下午 03:15	Chrome HTML D...	2 KB
index.html	2020/4/7 下午 03:15	Chrome HTML D...	7 KB
LICENSE	2020/4/7 下午 03:15	檔案	2 KB
README.md	2020/4/7 下午 03:15	MD 檔案	2 KB

但要注意他的 origin，若要推到其他 remote repo 記得修改

```
D:\GSS_project\roulette>git remote -v
origin D:/GSS_project/roulette.git (fetch)
origin D:/GSS_project/roulette.git (push)
```

3. Push

Push to Bare Repo

延續上方的 clone from local bare repo

```
$ cd roulette/
```

在 roulette 下新增一個 test.txt 檔案

(D:) > GSS_project > test > roulette

名稱	修改日期	類型	大小
.git	2020/4/9 下午 06:40	檔案資料夾	
test.txt	2020/4/9 下午 06:40	文字文件	1 KB

git add \ git commit

git push (to bare repo)

```
D:\GSS_project\roulette>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)

    test.txt

nothing added to commit but untracked files present (use "git add" to track)
D:\GSS_project\roulette>git add .
D:\GSS_project\roulette>git commit -m "test"
[master e07cdfd] test
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test.txt
D:\GSS_project\roulette>git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 233 bytes | 233.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0)
To D:/GSS_project/roulette.git
ea51lda..e07cdfd master -> master
```

進入一開始 clone 來源的 bare repo 查看

```
$ cd roulette.git/  
$ git log
```

搬遷 Remote Git Repo 情境

本地沒有完整專案

不需要 clone 一大包東西下來，可以 clone 一個 bare git repo (只有 git 儲存庫)，再推到新位址去就好

```
$ git clone --bare https://github.com/hbdoy/xxx  
$ cd xxx.git  
$ git push --mirror https://github.com/hbdoy/new_project
```