Git 和 GitHub簡易使用說明

大綱

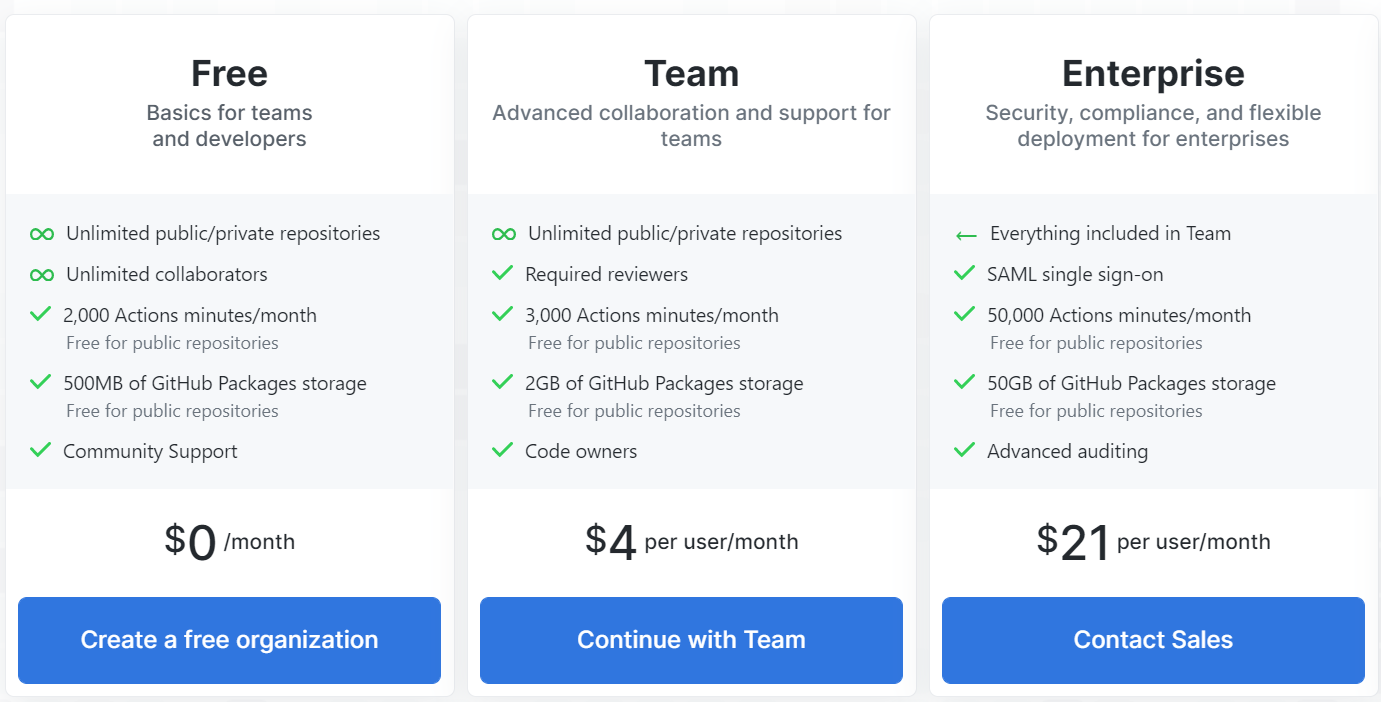
1. [Git 和 GitHub的簡介](#簡介)
2. [環境安裝與設定](#環境)
3. [創建新的Repository](#創建)
4. [將程式碼上傳github](#上傳)
5. [取消Commit](#取消)
6. [Git的Branch](#branch)
7. [看commit的內容](#內容)
8. [邀請別的帳號](#邀請)

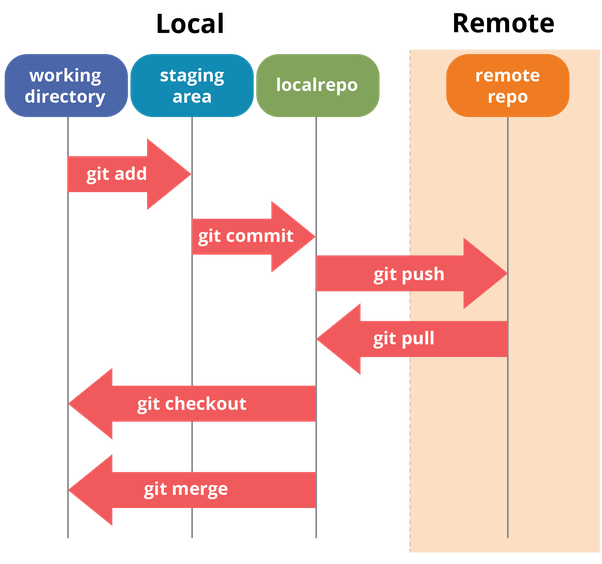
March 3, 2021

**一．****Git 和 GitHub的簡介**

[**Git**](https://zh.wikipedia.org/wiki/Git) 是一個分散式版本控制軟體，最初由 Linus Torvalds 創作（也是作業系統 [Linux](https://zh.wikipedia.org/zh-tw/Linux) 系統的開發者），其最初目的是為更好地管理 Linux kernel 開發而設計，其具備優秀的 merge tracing 合併程式碼的能力。

[**Github**](https://github.com/)則是一個支援 git 程式碼存取和遠端托管的平台服務，有許多的開放原始碼的專案都是使用 Github 進行程式碼的管理，2018年被微軟買下。





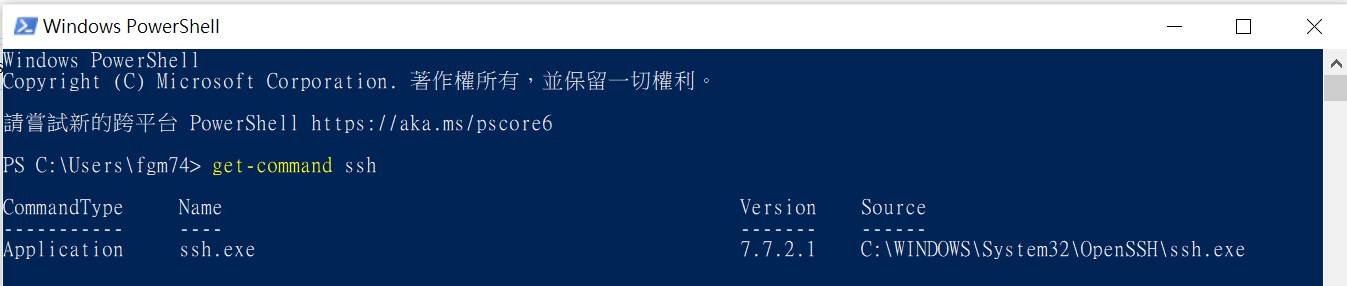
**二．****環境安裝與設定**

1. 安裝git for windows

<https://git-scm.com/download/win>

2. 用系統管理員開啟PowerShell，確認Windows 10有內建的SSH (1803的更新才有OpenSSH)

> Get-Command ssh



3. 把ssh-agent設定成manual啟動。

> Set-Service ssh-agent -StartupType Manual

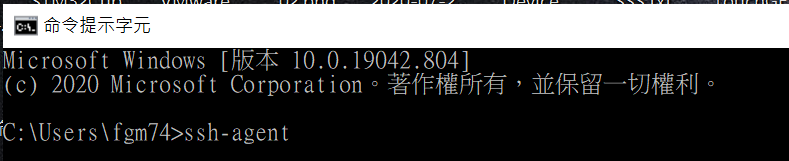


4. 開啟cmd視窗，設定git使用windows 10內建的OpenSSH

> git config --global core.sshCommand "'C:/Windows/System32/OpenSSH/ssh.exe'"

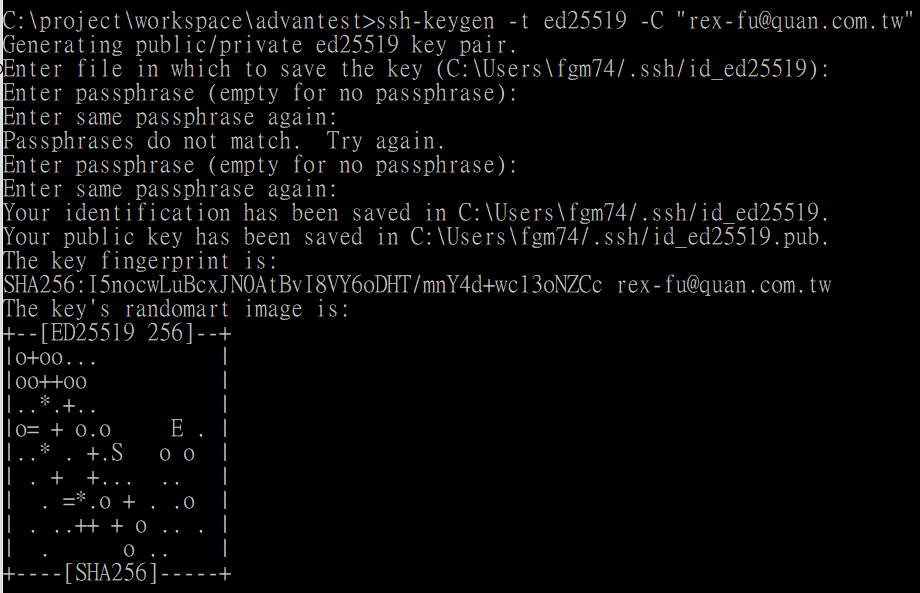


5. 啟動ssh-agent，每次重開機時，需重新啟動。



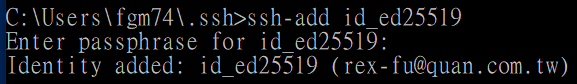
6. 產生ssh key，save的路徑直接按enter，密碼自行設定。

ssh-keygen -t ed25519 -C "*your\_email@example.com*"

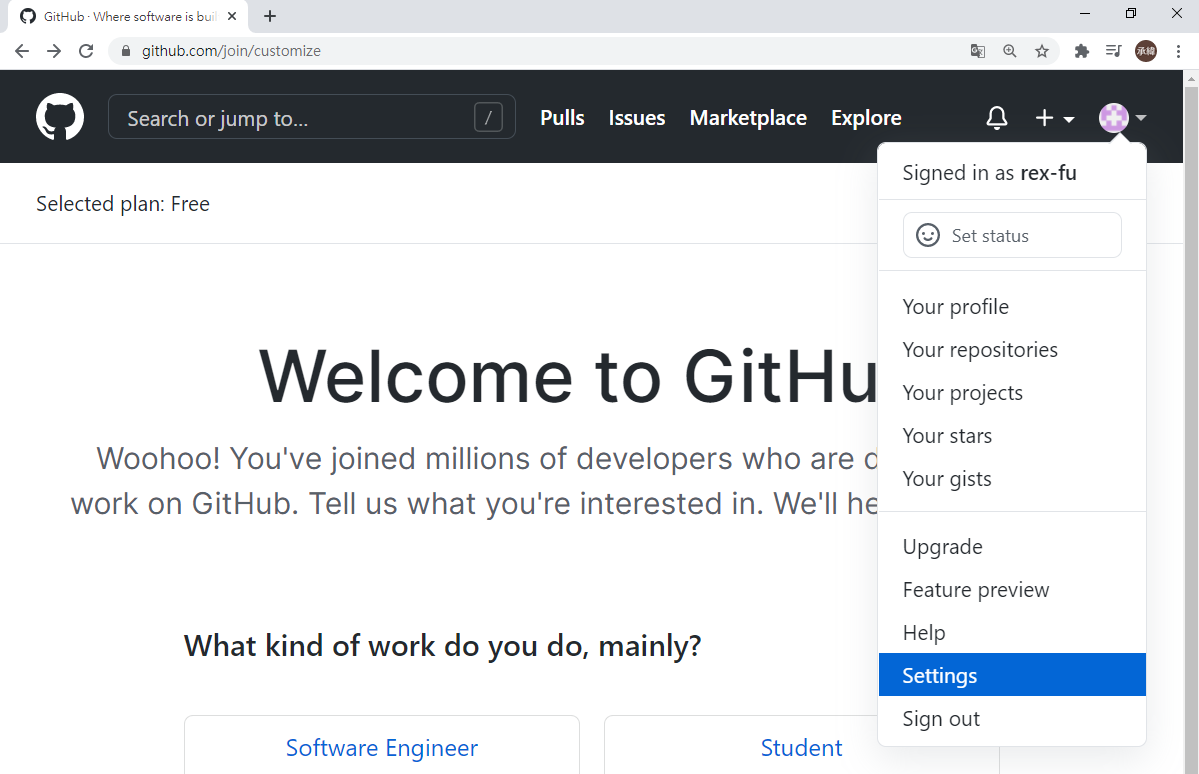


7. 到save的路徑，把key加到ssh-agent

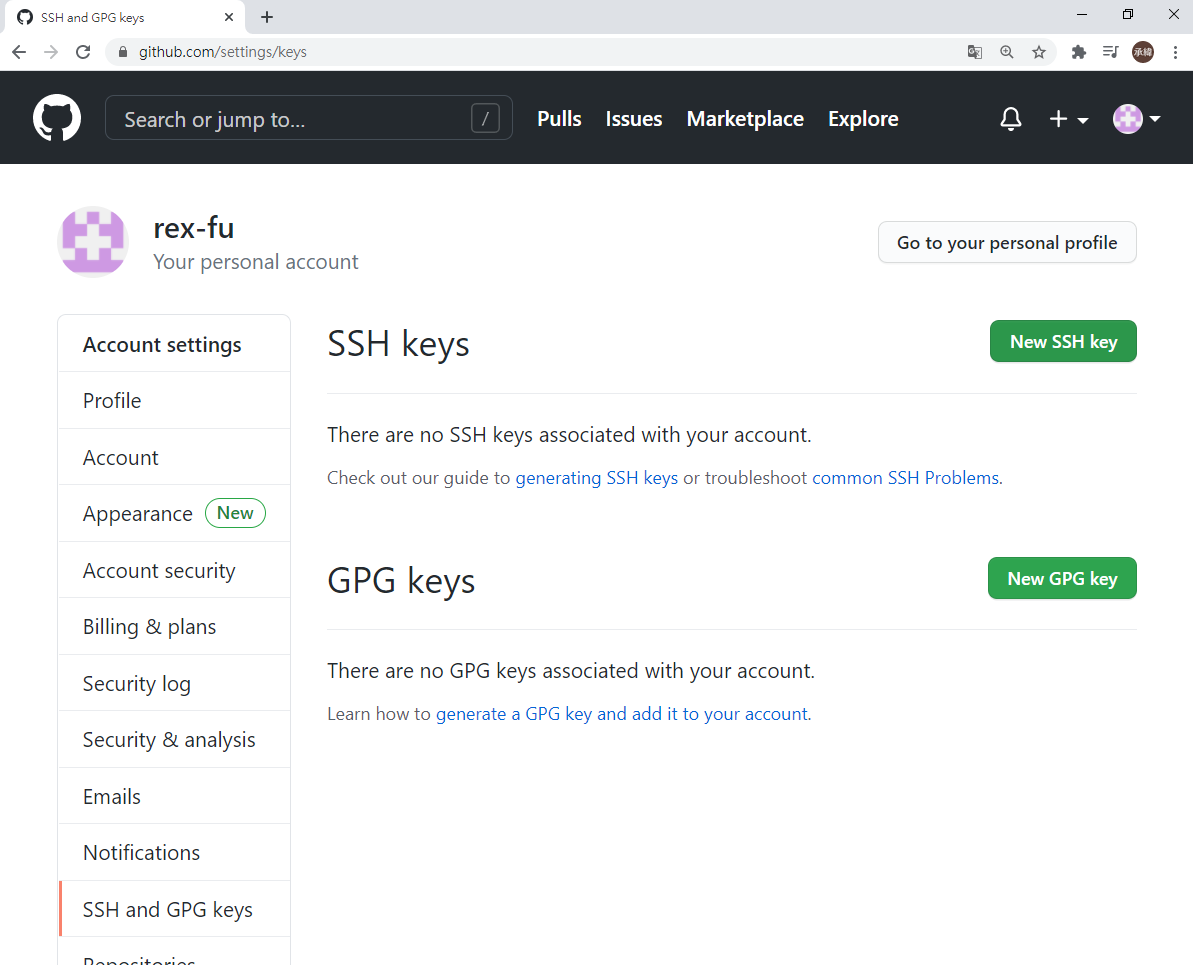
ssh-add id\_ed25519



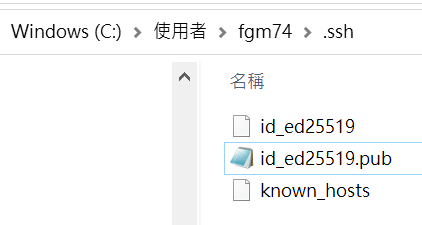
8. 先到github網站申請新的帳號，再到帳號底下的Settings

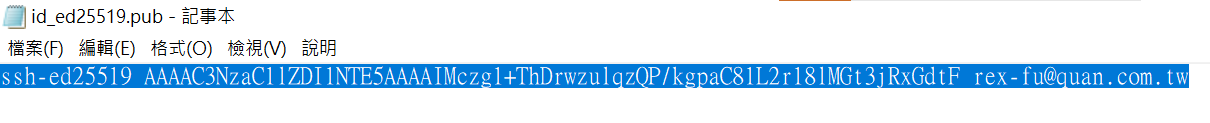


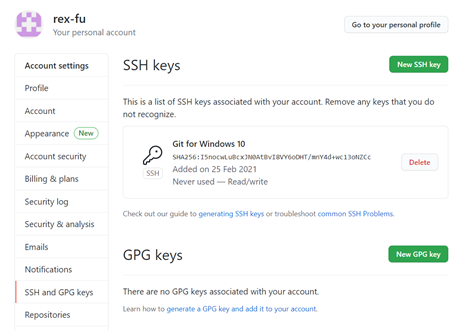
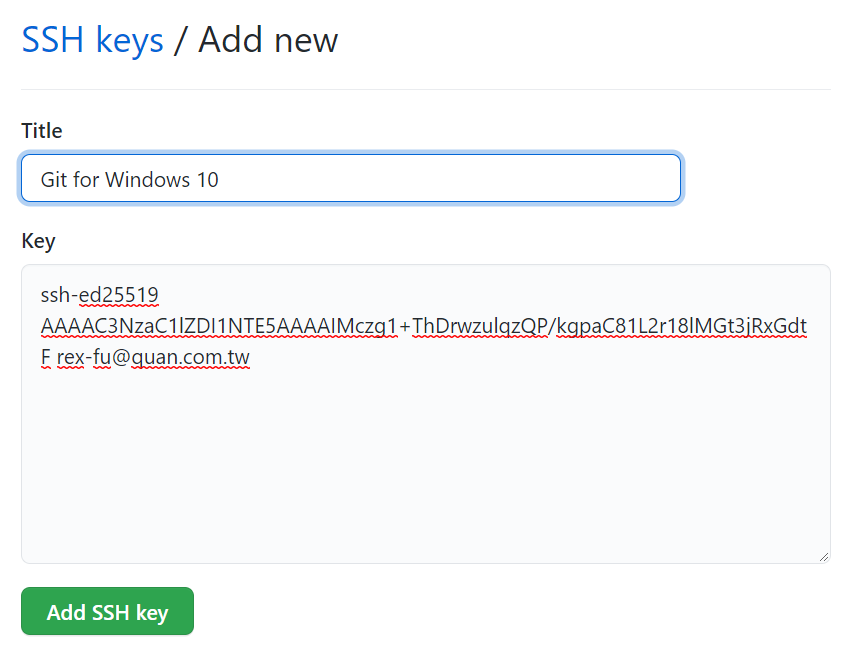
9. SSH and GPC keys，點選New SSH key



10. 將之前產生的id\_ed25519.pub的檔案內容，複製貼到上面。



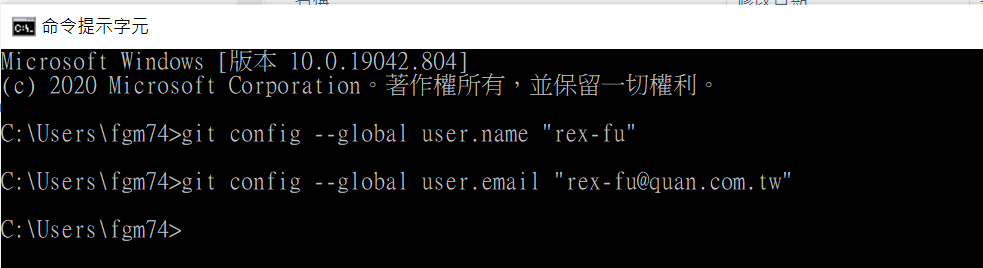




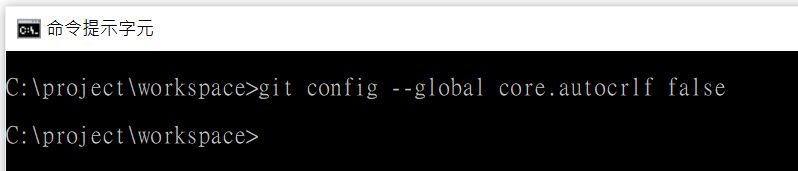
11. 開啟cmd視窗，設定git的user name和email

git config --global user.name "your name"

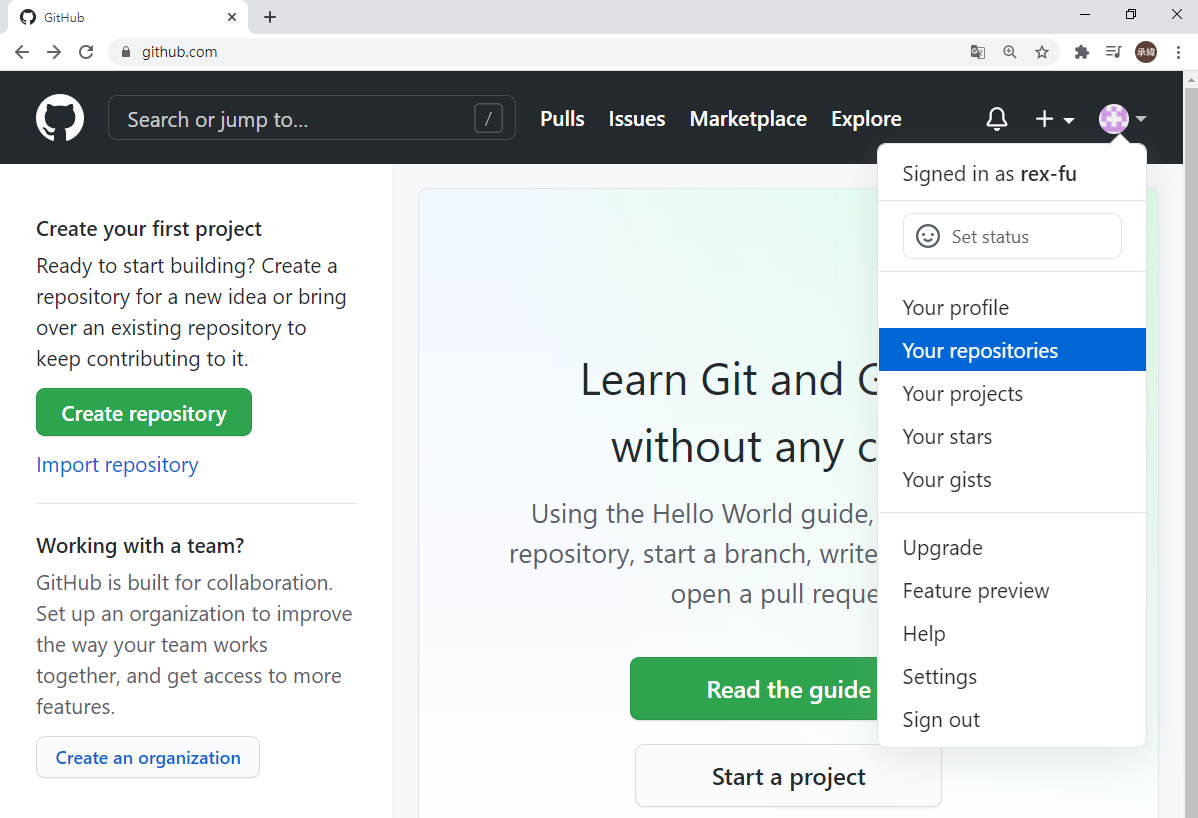
git config --global user.email "email@example.com"



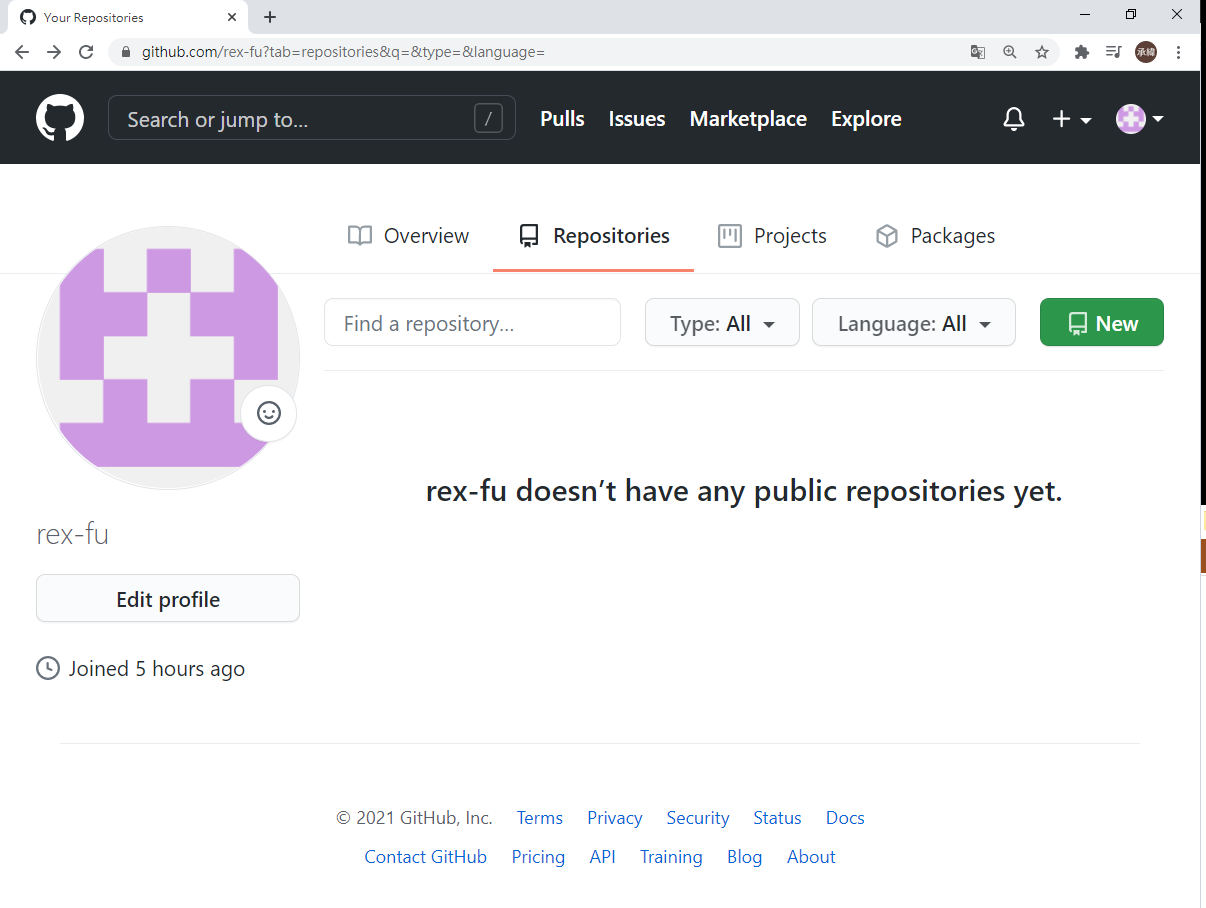
12. 因為案子的程式碼為Windows環境下使用的，所以關掉自動CRLF endings的校正。



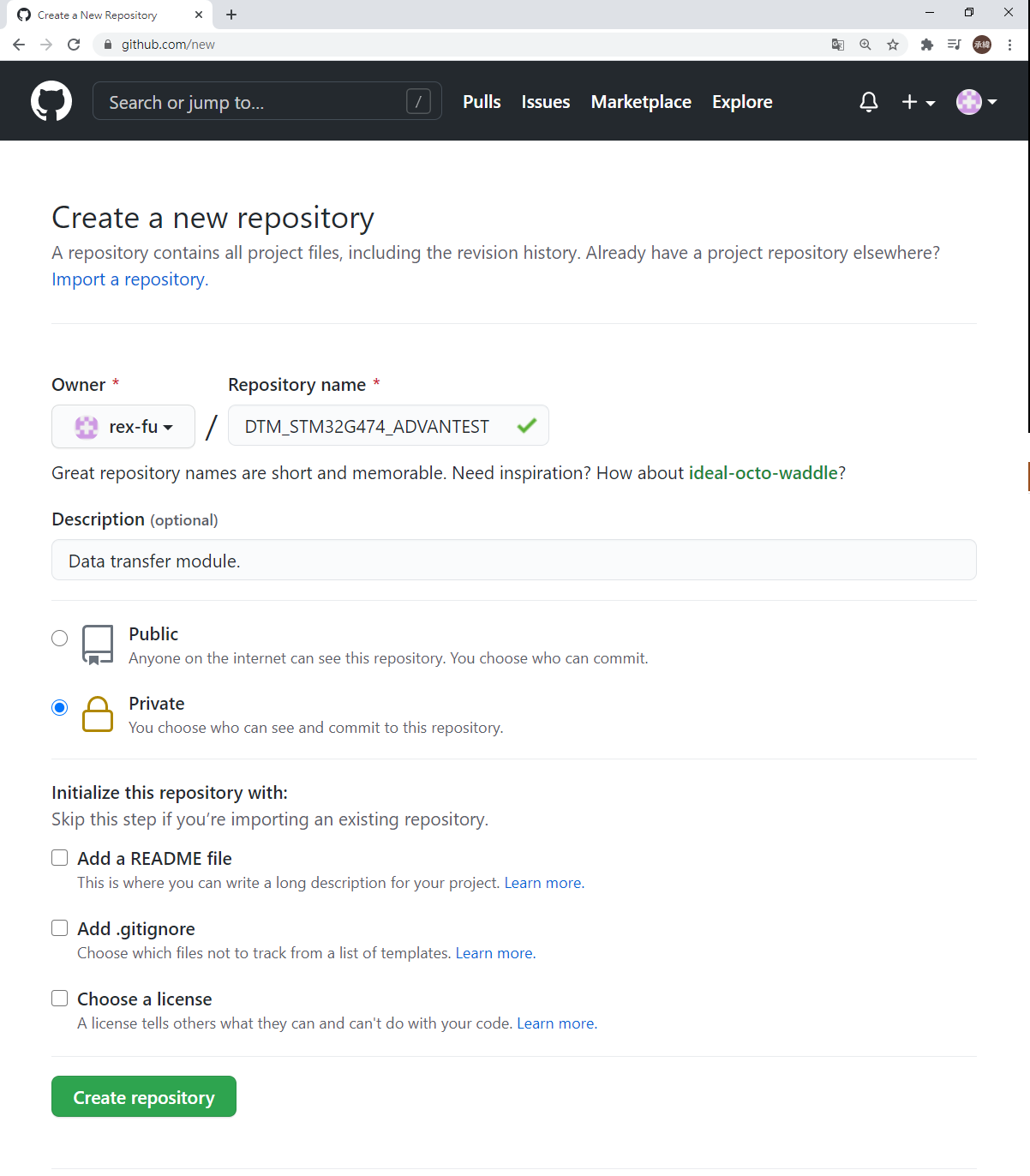
1. **創建新的Repository**
2. 再到帳號底下的Your repositories



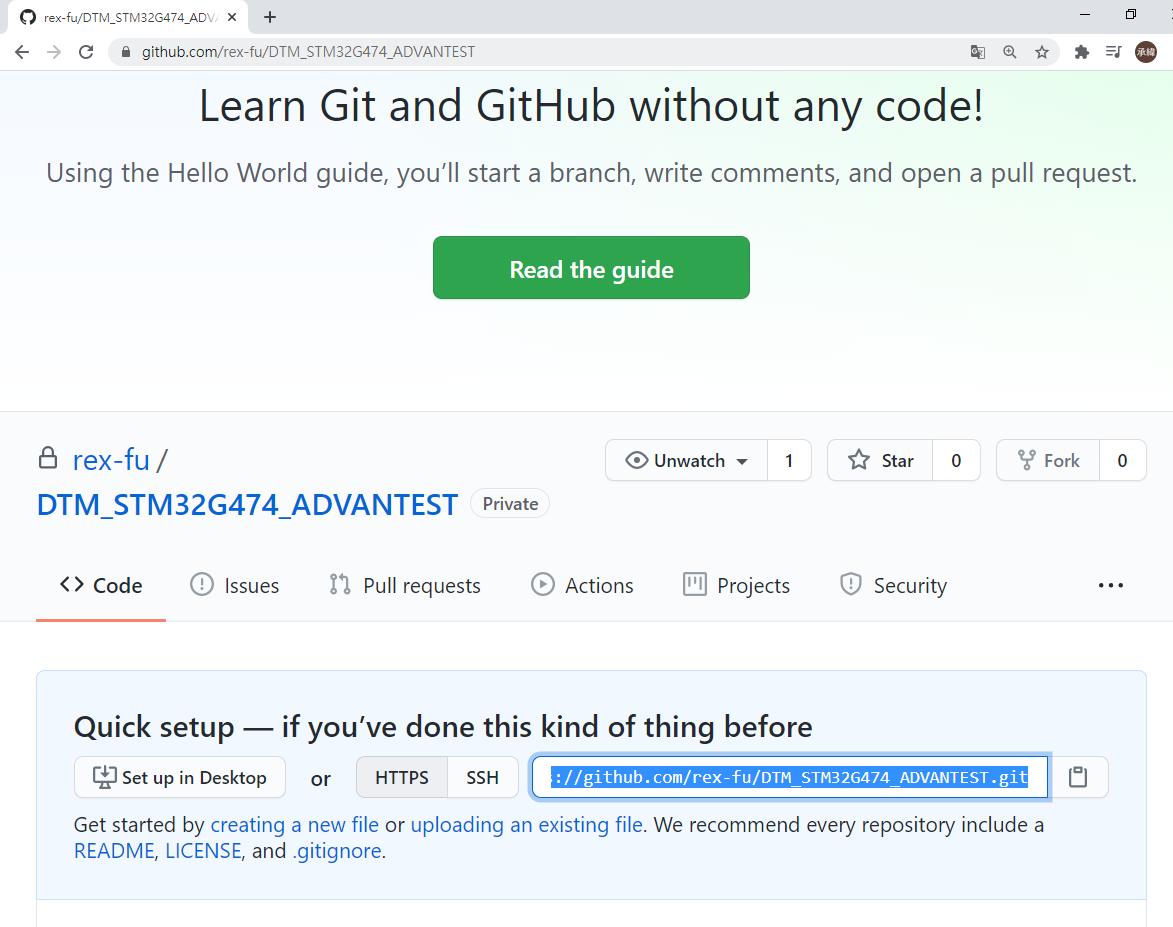
1. 點選New



1. 命名repository的name後，勾選Private，Private只有個人和你邀請的帳號，可以使用這個repository。

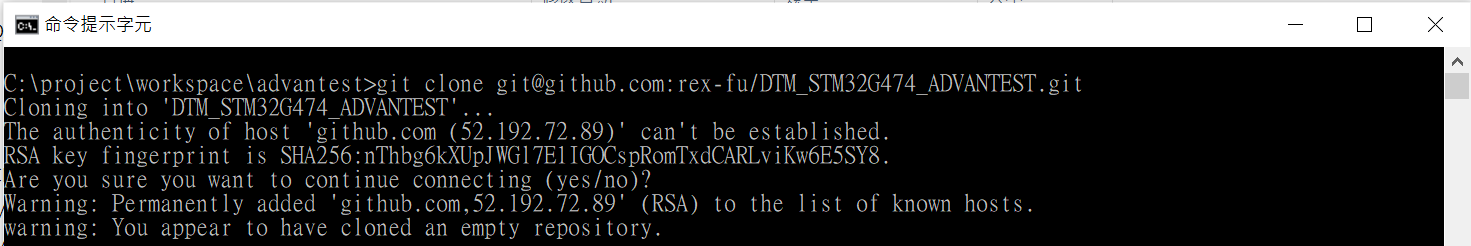


1. 把新的repository資訊，同步到筆電
2. 在資料庫內，點選SSH，複製右邊的字串



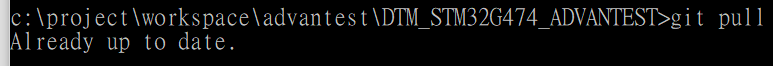
1. 在筆電新增一個資料夾

> git clone 複製的字串



**四．****將程式碼上傳github**

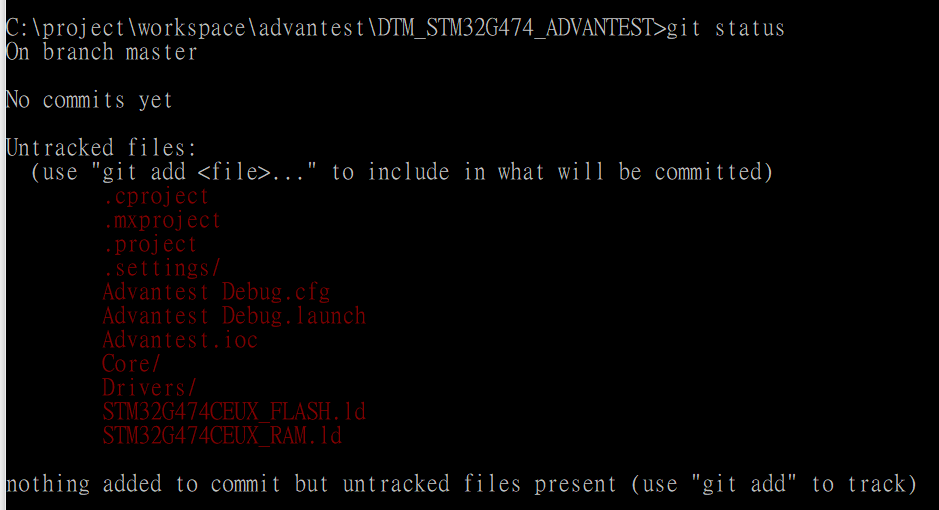
1. 先下git pull，跟github做一次同步，每次上code前一定要做。



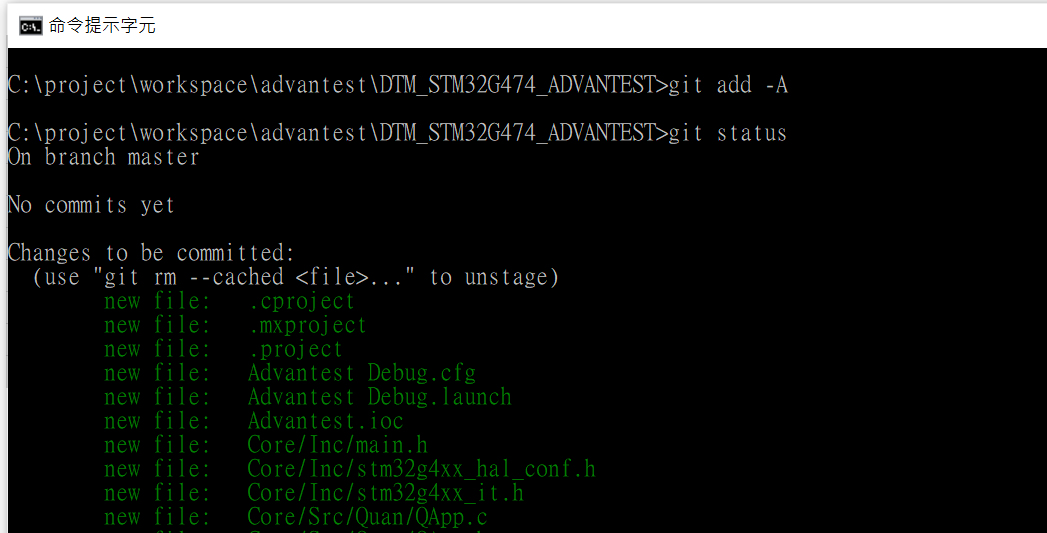
1. 將要上傳的程式碼，複製到資料夾



1. 下git status，紅色代表資料夾內，有變動但尚未到暫存區(stage)的檔案。

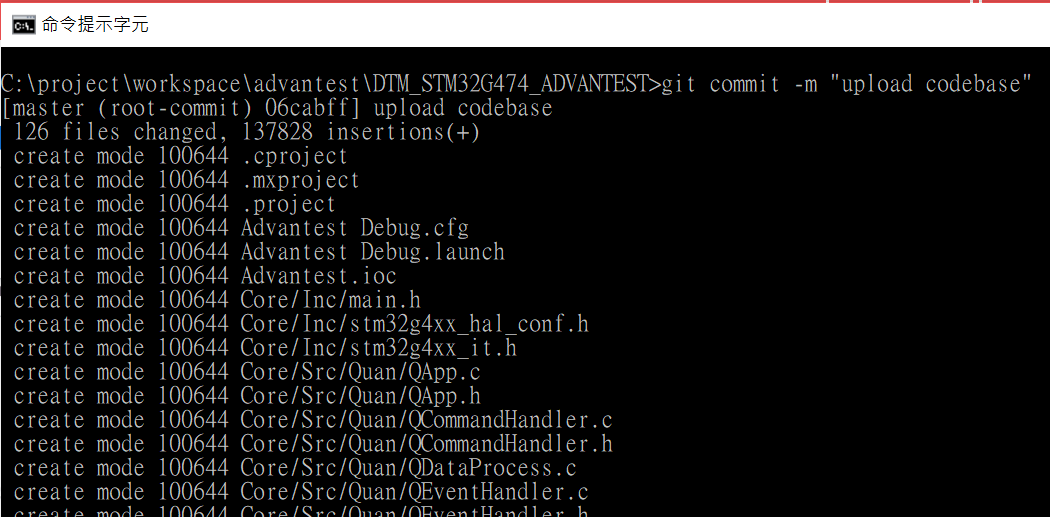


1. 下git add -A，把變動的檔案加入stage，再下git status，可看到變成綠色。

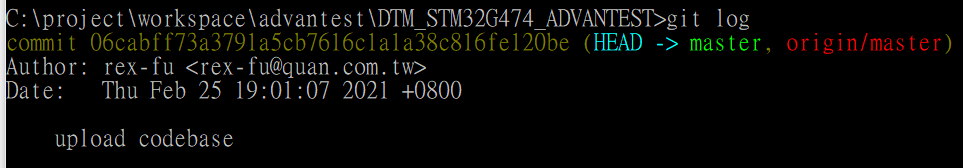


1. 將stage的檔案commit到repository

> git commit -m “這次提交的敘述”

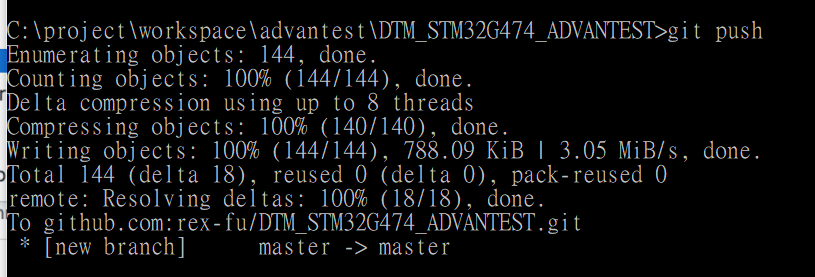


1. 下git log看資料庫的commit紀錄。

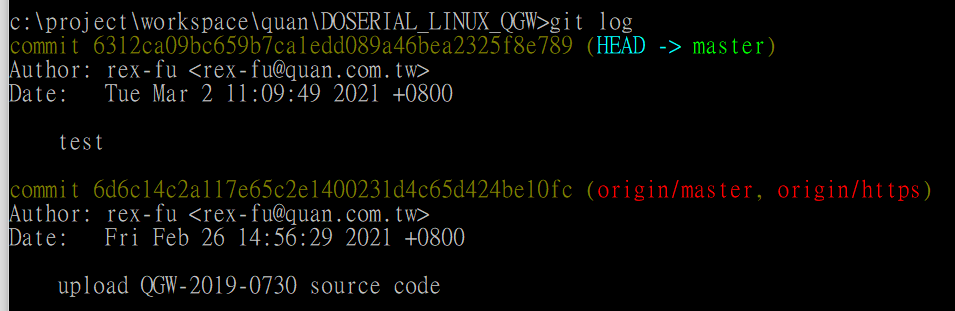


1. 將筆電repository的更新，push到github

> git push



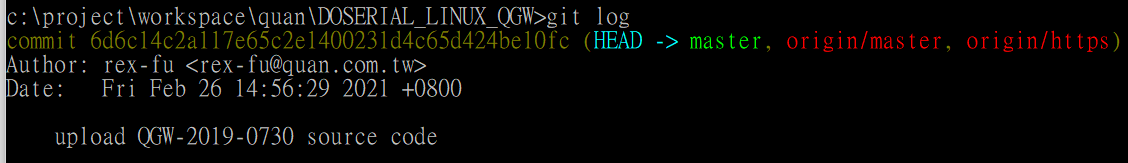
1. **取消Commit**
2. 狀況一: 如果尚未上傳github
3. 例如說想把最新的commit “test”取消



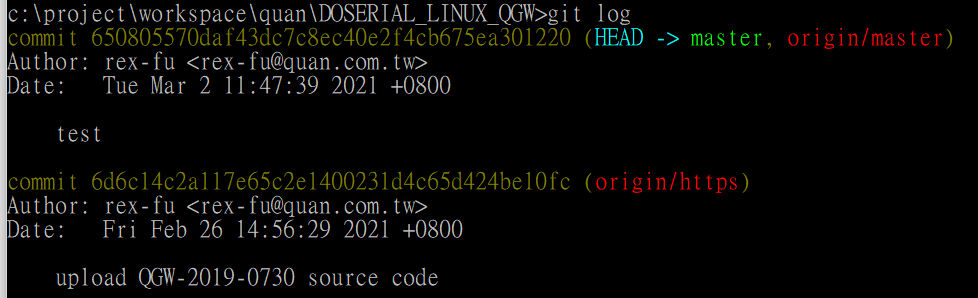
1. git reset --hard HEAD~1 (--hard:會刪除檔案，--soft:不會刪除資料夾檔案，而stage還會存在更改，--mixed:不會刪除檔案，stage也不會存在更改)



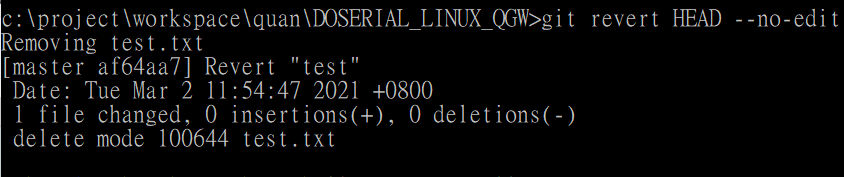
1. git log，可發現已消除commit。



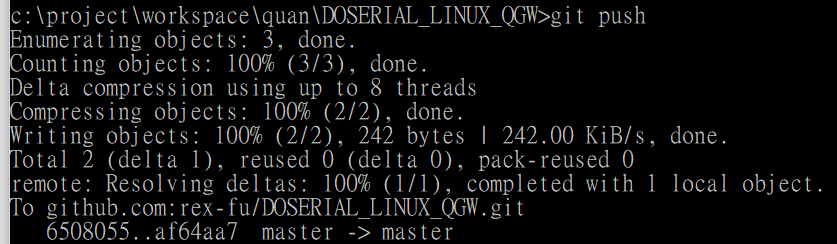
1. 狀況二: 如果已上傳github
2. git log



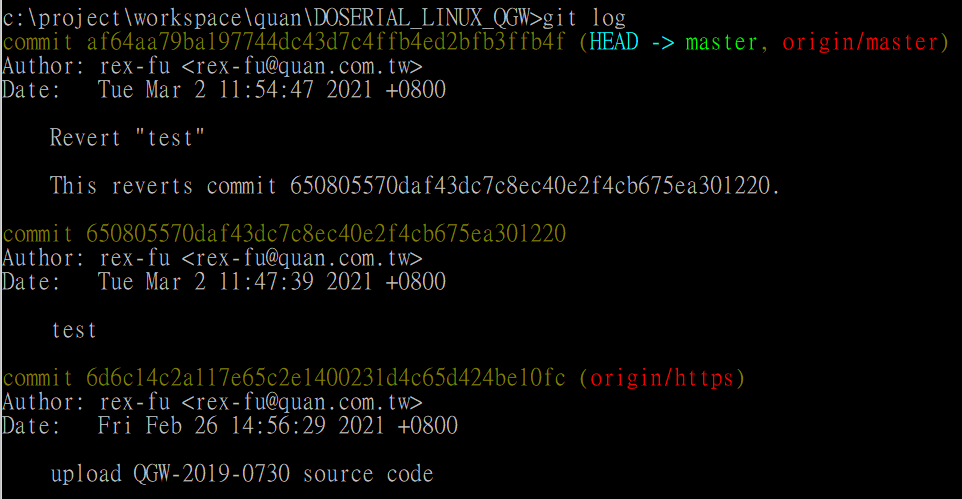
1. git revert HEAD –no-edit



1. git push到github



1. git log，會發現多了一個revert的提交紀錄



1. **Git的Branch**



1. 創建branch

> git branch 分支名稱



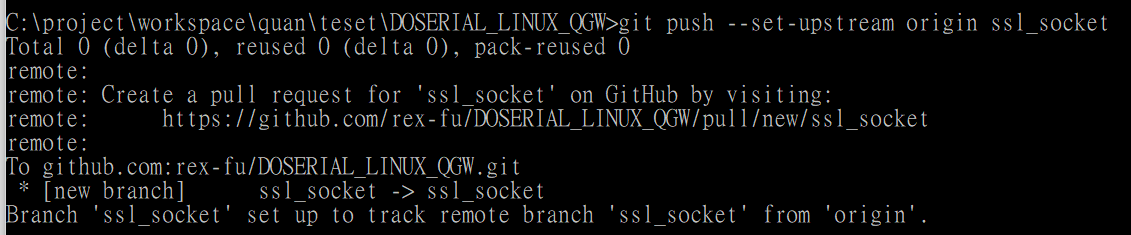
1. 切換新的branch

> git checkout 分支名稱

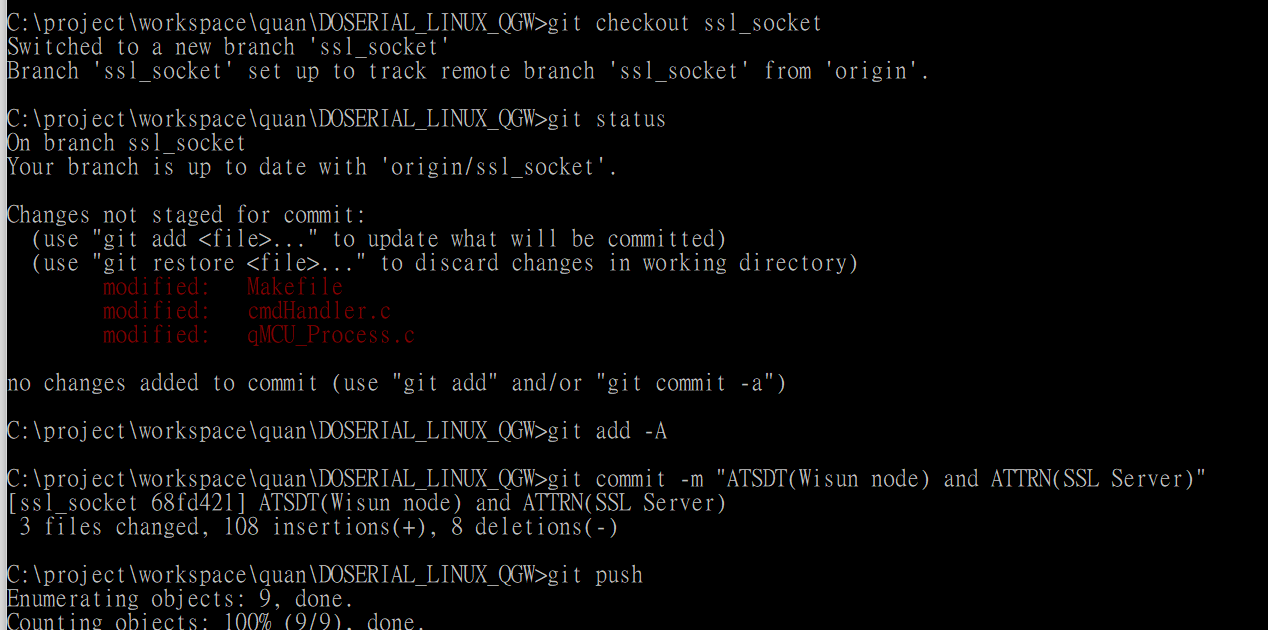


1. 將branch，同步到github

> git push –set-upstream origin 分支名稱

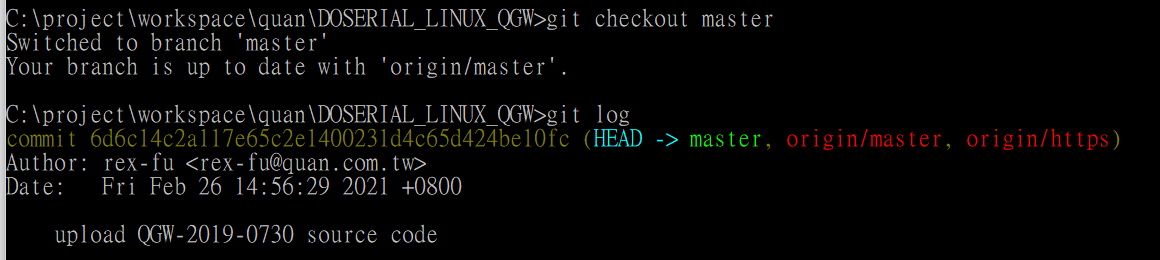


1. 在分支上code
2. git checkout 分支名稱
3. 檢查是否有切換成功，git status
4. git add -A
5. git commit -m “提交敘述”
6. git push



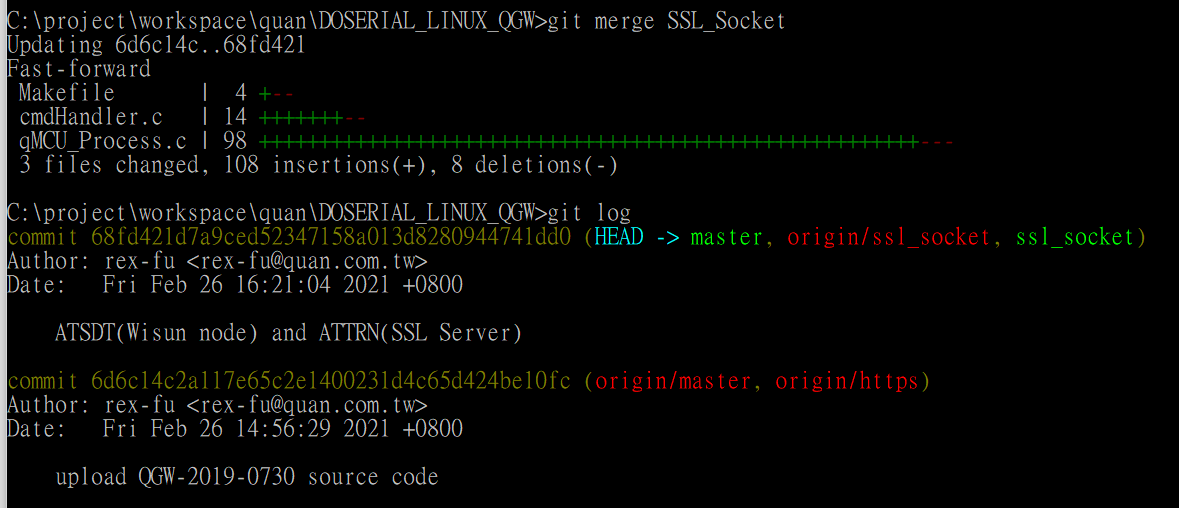
1. 分支的commit，整合回master
2. 切回master

> git checkout master

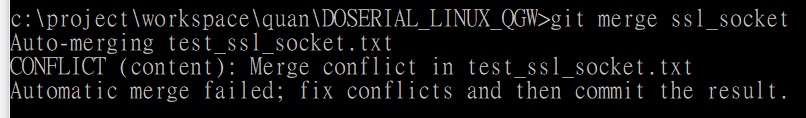


b. Merge分支的到master

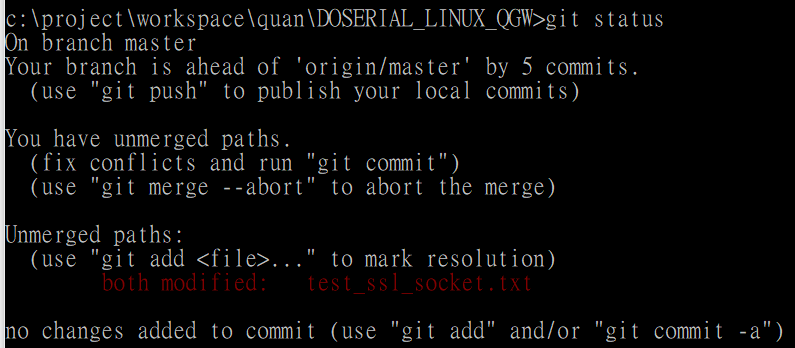
>git merge 分支名稱



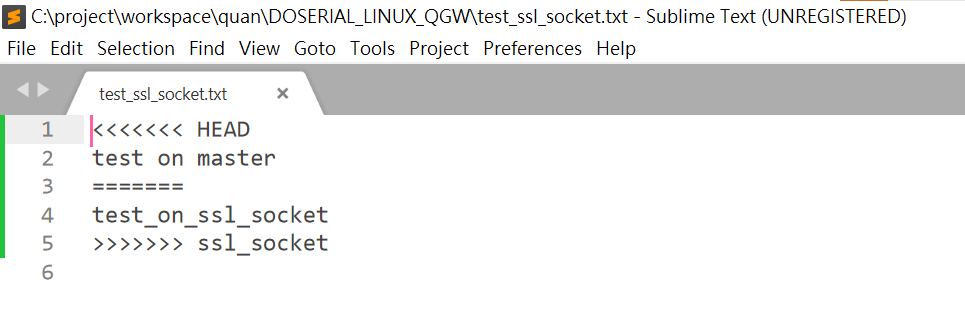
1. 如果Merge時發生衝突



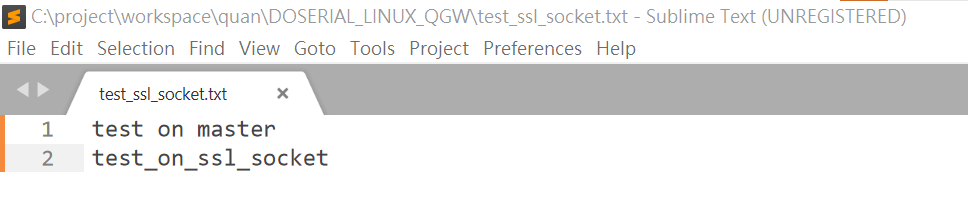
1. git status看哪個檔案是both modified



1. 打開那個檔案



1. 修改檔案，並存檔



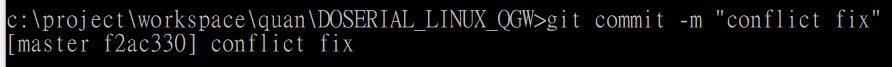
1. 把修改後的檔案加入stage

> git add -A

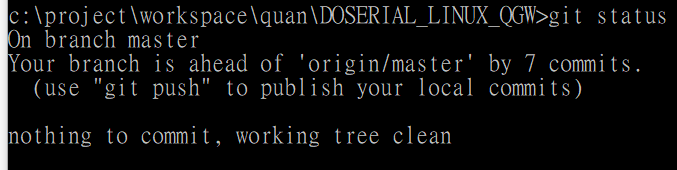


1. 提交

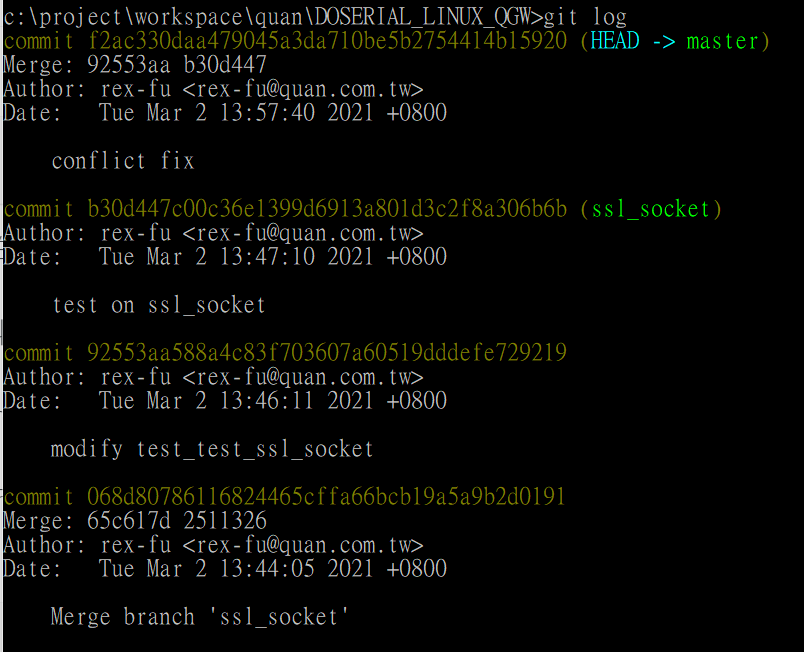
> git commit -m “conflict fix”



1. git status

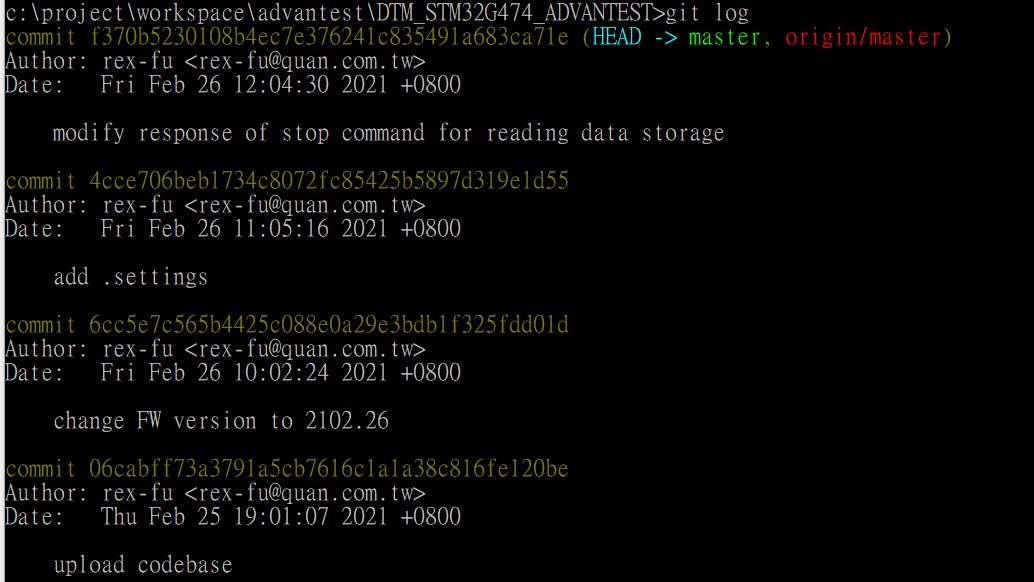


1. git log可以看到分支的提交紀錄



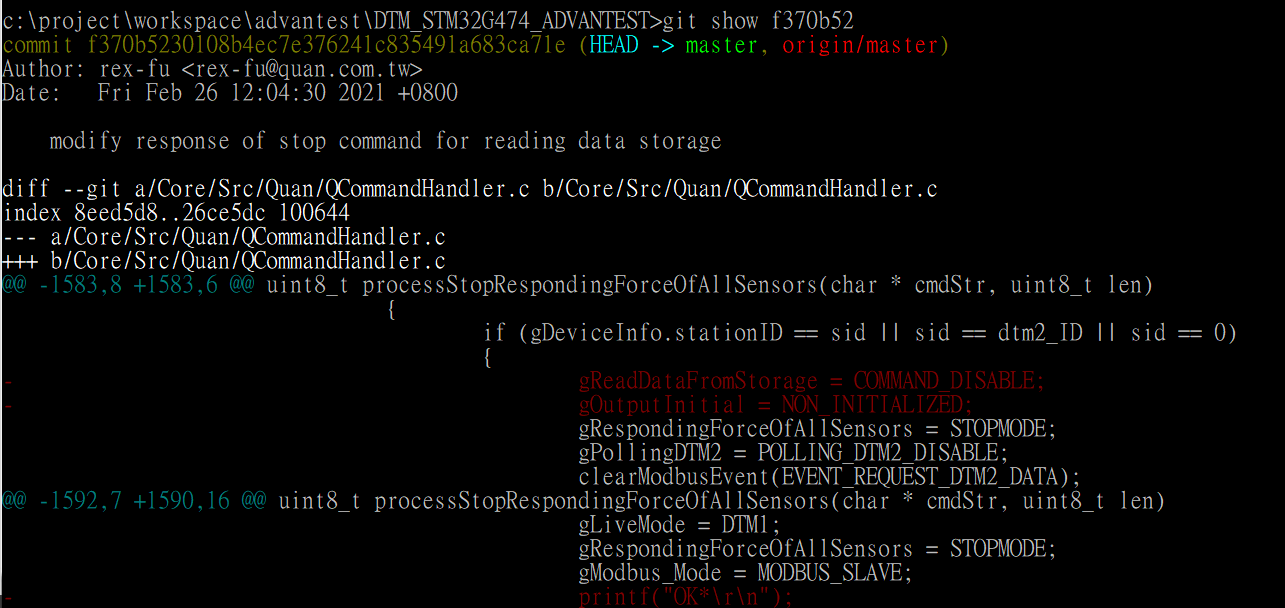
1. **看commit的內容**
2. 先看commit紀錄

> git log

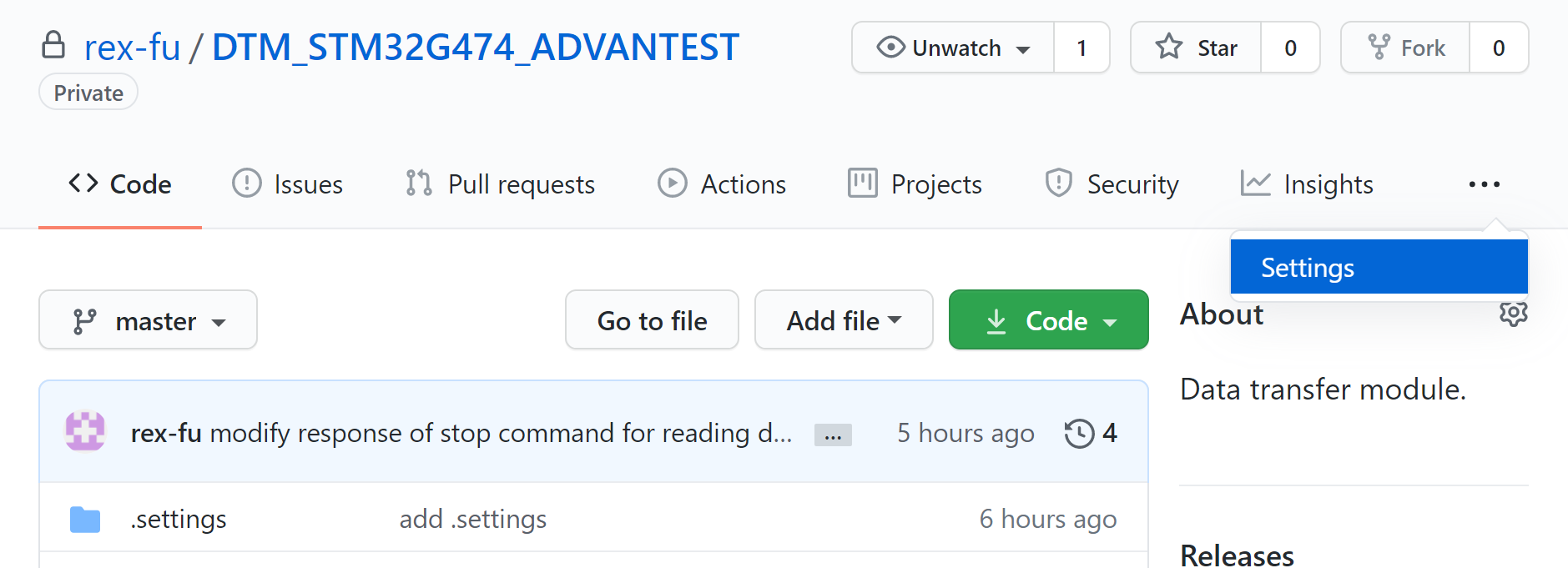


1. 看指定commit所更新的內容

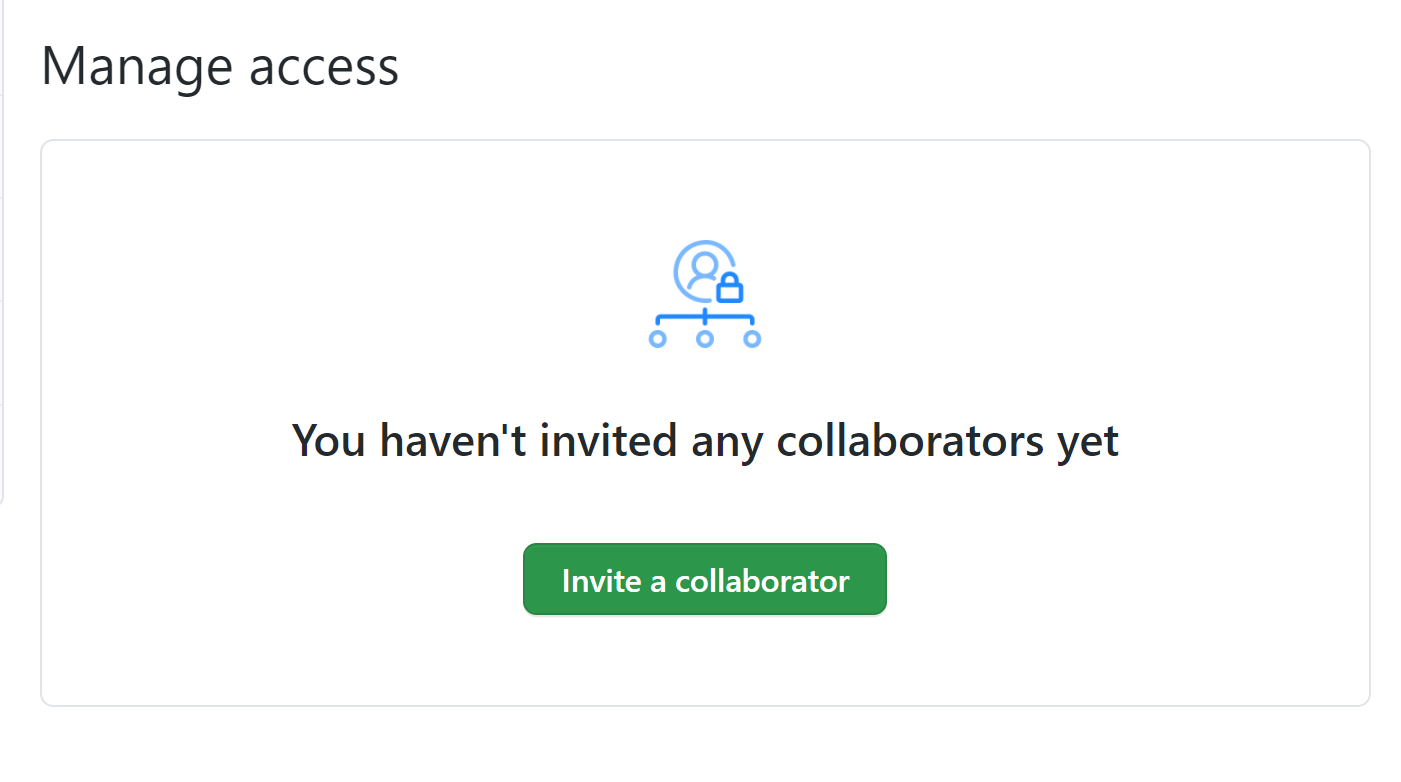
> git show 號碼



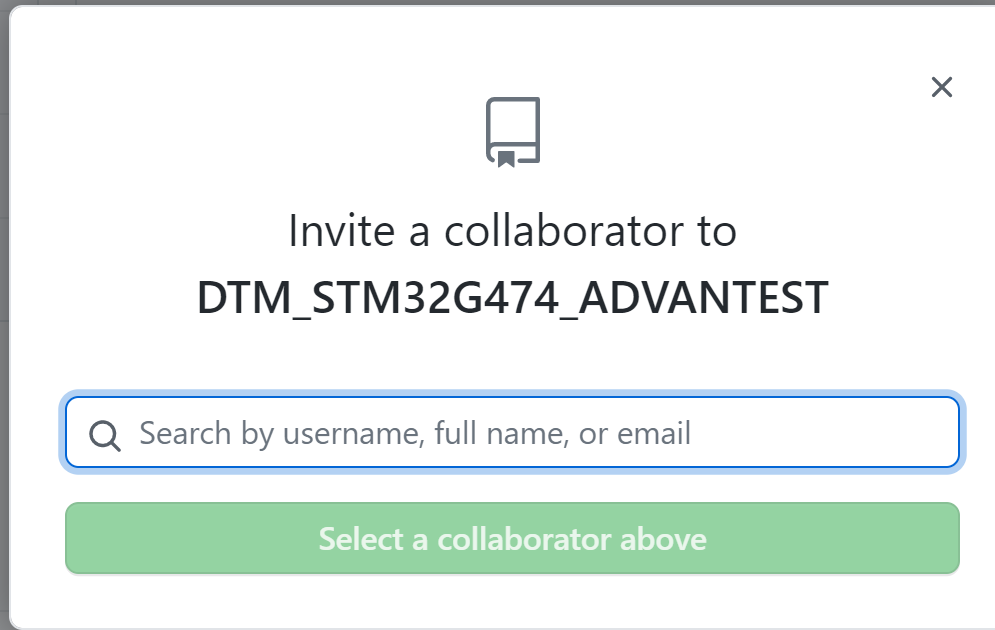
1. **邀請別的帳號**
2. 到repository下，選擇Settings



1. 再選Manage access下的Invite a collaborator



1. 輸入對方的資訊



1. 對方就會收到邀請的e-mail

