

Git Lab 2

一、本節目地

- GitHub 基本指令的使用及介紹

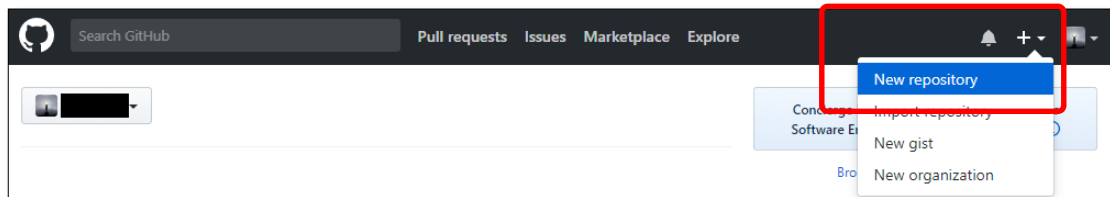
二、設計重點

- 在本地端已經完成專案的初始化及檔案架構後，本次練習是要將建構好的專案推送到遠端伺服器上

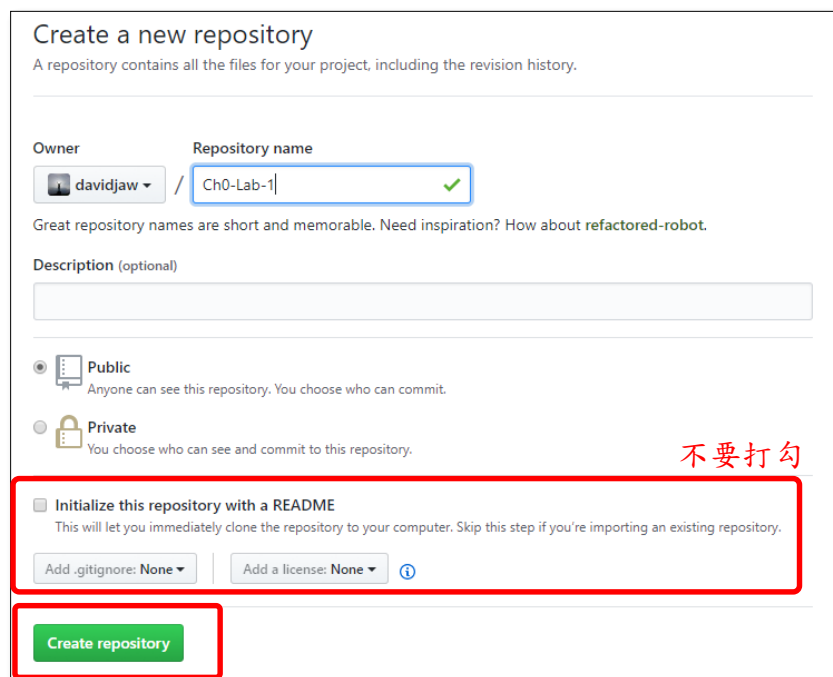
三、設計步驟

1. 於 GitHub 上開設新的 Repository

點開網站上方的+號，並點選第一項



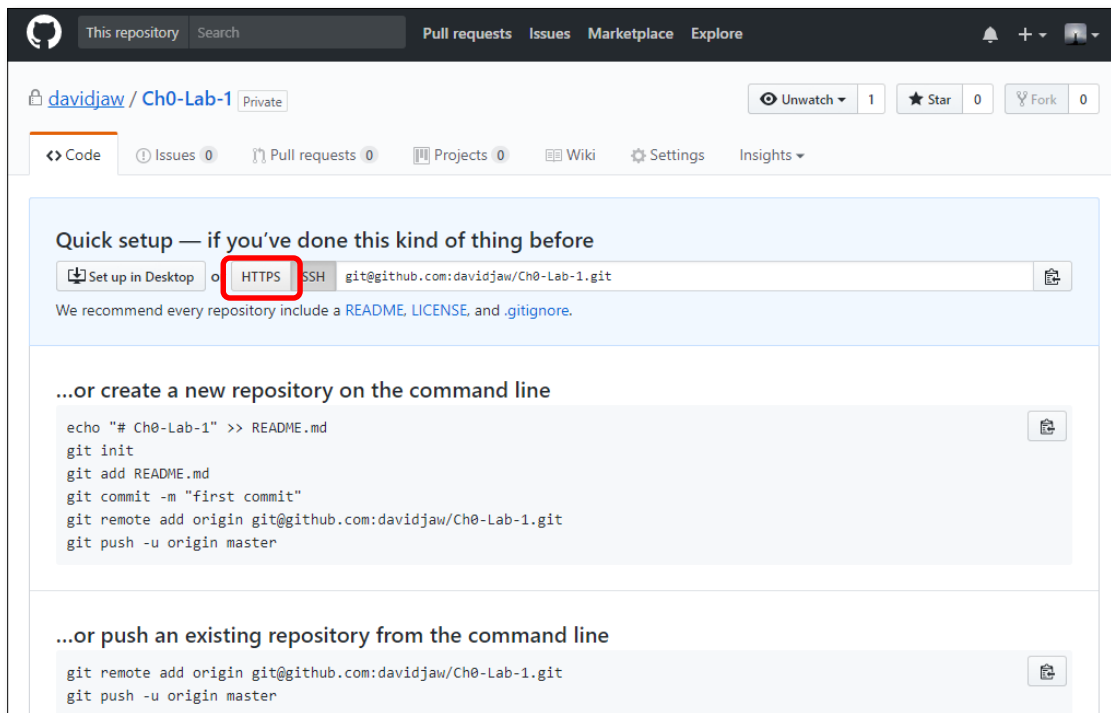
2. 在 Repository name 欄位輸入 Ch0-Lab-1



不要打勾

名稱輸入完畢後點選創建

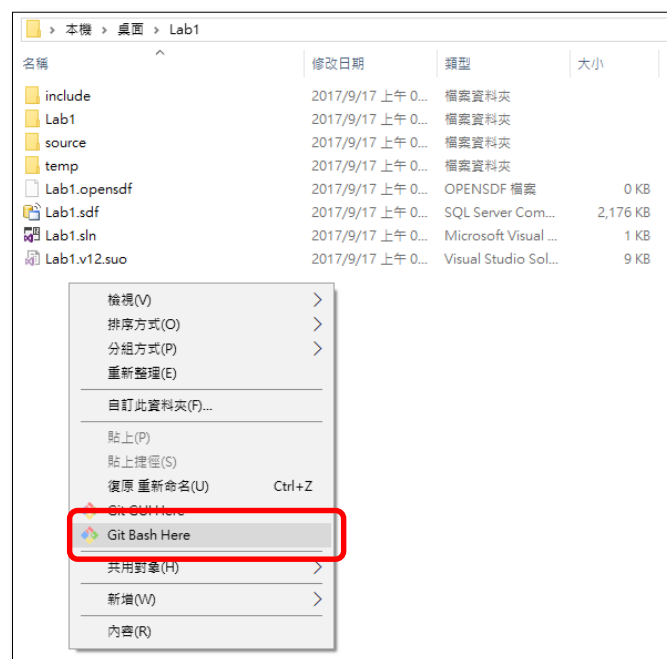
3. 創建成功應該會到以下畫面，點選畫面中的 **HTTPS**



4. 點選完畢後網址會變成 https 開頭，請複製該欄位內文字(或點選紅框處)



5. 開啟專案資料夾，空白處按下右鍵選擇 **Git Bash Here**



6. 在視窗內輸入 `git init` 來在專案資料夾初始化本地資料庫

```
Jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1
$ git init
Initialized empty Git repository in C:/Users/Jdway/Desktop/Lab1/.git/

Jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1 (master)
$ |
```

成功的話會在專案資料夾中多出一個 `.git` 的隱藏資料夾



名稱	修改日期	類型	大小
include	2017/9/17 上午 0...	檔案資料夾	
Lab1	2017/9/17 上午 0...	檔案資料夾	
source	2017/9/17 上午 0...	檔案資料夾	
temp	2017/9/17 上午 0...	檔案資料夾	
Lab1.opensdf	2017/9/17 上午 0...	OPENSDF 檔案	0 KB
Lab1.sdf	2017/9/17 上午 0...	SQL Server Com...	2,176 KB
Lab1.sln	2017/9/17 上午 0...	Microsoft Visual ...	1 KB
Lab1.v12.suo	2017/9/17 上午 0...	Visual Studio Sol...	9 KB
.git	2017/9/17 上午 0...	檔案資料夾	

7. 使用 `git ststus` 以觀察目前檔案狀況

```
Jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1 (master)
$ git status
On branch master

No commits yet

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

    Lab1.opensdf
    Lab1.sdf
    Lab1.sln
    Lab1.v12.suo
    Lab1/
    source/
    temp/

nothing added to commit but untracked files present (use "git add" to track)
```

發現目前的修改都沒有被列入到 commit 佇列中

8. 使用 `git add .` 來將所有變更都加入到 commit 佇列

注意：若沒有把 visual studio 關閉，會出現以下訊息

```
jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1 (master)
$ git add .
error: open("Lab1.opensdf"): Permission denied
error: unable to index file Lab1.opensdf
fatal: adding files failed

jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1 (master)
$ |
```

關閉後使用：

```
jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1 (master)
$ git add .
warning: CRLF will be replaced by LF in Lab1.sln.
The file will have its original line endings in your working directory.
warning: CRLF will be replaced by LF in Lab1/Lab1.vcxproj.
The file will have its original line endings in your working directory.
warning: CRLF will be replaced by LF in Lab1/Lab1.vcxproj.filters.
The file will have its original line endings in your working directory.
warning: CRLF will be replaced by LF in Lab1/Lab1.vcxproj.user.
The file will have its original line endings in your working directory.
warning: CRLF will be replaced by LF in source/main.c.
The file will have its original line endings in your working directory.
warning: CRLF will be replaced by LF in temp/debug/Lab1.log.
The file will have its original line endings in your working directory.
warning: CRLF will be replaced by LF in temp/debug/Lab1.tlog/Lab1.lastbuildstate.
The file will have its original line endings in your working directory.

jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1 (master)
$ |
```

(這些警告是因為助教的換行設定跟 Git 設定的不同，不必理會)

9. 使用 `git status` 確認目前狀態

```
jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   Lab1.sdf
        new file:   Lab1.sln
        new file:   Lab1.v12.suo
        new file:   Lab1/Lab1.vcxproj
        new file:   Lab1/Lab1.vcxproj.filters
        new file:   Lab1/Lab1.vcxproj.user
        new file:   source/main.c
        new file:   temp/debug/Lab1.exe
        new file:   temp/debug/Lab1.ilc
        new file:   temp/debug/Lab1.log
        new file:   temp/debug/Lab1.pdb
        new file:   temp/debug/Lab1.tlog/CL.read.1.tlog
        new file:   temp/debug/Lab1.tlog/CL.write.1.tlog
        new file:   temp/debug/Lab1.tlog/Lab1.lastbuildstate
        new file:   temp/debug/Lab1.tlog/cl.command.1.tlog
        new file:   temp/debug/Lab1.tlog/link.command.1.tlog
        new file:   temp/debug/Lab1.tlog/link.read.1.tlog
        new file:   temp/debug/Lab1.tlog/link.write.1.tlog
        new file:   temp/debug/main.obj
        new file:   temp/debug/vc120.idb
        new file:   temp/debug/vc120.pdb

jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1 (master)
$ |
```

如上圖，所有檔案都被加入到 commit 佇列中

10. 使用 git commit 來提交一次檔案更動到本地資料庫中

```
jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1 (master)
$ git commit -m "Complete Lab1"
[master (root-commit) 29c6bed] Complete Lab1
21 files changed, 145 insertions(+)
create mode 100644 Lab1.sdf
create mode 100644 Lab1.sln
create mode 100644 Lab1.v12.suo
create mode 100644 Lab1/Lab1.vcxproj
create mode 100644 Lab1/Lab1.vcxproj.filters
create mode 100644 Lab1/Lab1.vcxproj.user
create mode 100644 source/main.c
create mode 100644 temp/debug/Lab1.exe
create mode 100644 temp/debug/Lab1.ilc
create mode 100644 temp/debug/Lab1.log
create mode 100644 temp/debug/Lab1.pdb
create mode 100644 temp/debug/Lab1.tlog/CL.read.1.tlog
create mode 100644 temp/debug/Lab1.tlog/CL.write.1.tlog
create mode 100644 temp/debug/Lab1.tlog/Lab1.lastbuildstate
create mode 100644 temp/debug/Lab1.tlog/cl.command.1.tlog
create mode 100644 temp/debug/Lab1.tlog/link.command.1.tlog
create mode 100644 temp/debug/Lab1.tlog/link.read.1.tlog
create mode 100644 temp/debug/Lab1.tlog/link.write.1.tlog
create mode 100644 temp/debug/main.obj
create mode 100644 temp/debug/vc120.idb
create mode 100644 temp/debug/vc120.pdb
jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1 (master)
$ |
```

語法：git commit -m “<提交訊息>”

將紅字部分替換成提交訊息即可

11. 使用 git push 將本地資料庫同步到遠端資料庫上

直接使用 git push 會發現我們尚未設定遠端資料庫的位置：

```
jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1 (master)
$ git push
fatal: No configured push destination.
Either specify the URL from the command-line or configure a remote repository using

    git remote add <name> <url>

and then push using the remote name

    git push <name>

jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1 (master)
$ |
```

因此，我們將步驟四複製的那串網址設定為遠端伺服器：

```
jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1 (master)
$ git remote add origin https://github.com/davidjaw/Ch0-Lab-1.git
jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1 (master)
$ |
```

語法：git remote add origin <url>

將紅字部分替換成複製的網址

12. 使用 `git push` 將本地資料庫同步到遠端資料庫上

由於尚未設定本地的 branch 與遠端 branch 的關聯，在第一次使用 `git push` 時必須如以下語法：

```
git push -u origin master
```

```
Jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1 (master)
$ git push -u origin master
Counting objects: 28, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (26/26), done.
Writing objects: 100% (28/28), 653.77 KiB | 7.18 MiB/s, done.
Total 28 (delta 5), reused 0 (delta 0)
remote: Resolving deltas: 100% (5/5), done.
To https://github.com/davidjaw/Ch0-Lab-1.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.

Jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop/Lab1 (master)
$ |
```

之後如果要再做提交程就只需要 `git push` 即可。

四、檢查重點

上傳成功後，GitHub 網頁應該會如下圖

