**兩個用戶帳號登入Github on Windows**

**通常在你已經有一個Github帳號下的操作並不會有這樣的困擾, 但如果你有工作上的需要會有另一個工作上的Github帳號, 在同時有兩個Github帳號下, 指令操作git pul, git push就會發生非常困擾及混亂的情形, 以下是解決方式**

**※本地用戶端無法快速切換登入Github倉庫, 本地端用Git Bash解決問題**

在Github同時有私人帳號&工作帳號, 在Windows下切換方式將採用ssh key會比較方便. (私人帳號產生的ssh key為default account)

即在用戶端建立生成兩個ssh key, 將ssh key公鑰, 複製新增到Github.com->Setting->SSH and GPG keys (Add 加入這兩個公鑰ssh key)

**※用Git bash在用戶端(Windows)創建工作帳號ssh key**

Git bash登入在準備克隆github repo.的目錄下

$ ssh-keygen -t rsa -C "帳號"

or

$ ssh-keygen -t rsa –b 4096 -C "帳號"

or

$ ssh-keygen -t rsa -b 4096 -f $HOME/.ssh/id\_rsa -P "test1234" -C "帳號"

(過程會有要求輸入郵箱密碼, 如果不輸入密碼表示私鑰並不加密碼輸入認證, 密碼與帳號沒有關係是本地端多一層保護的密碼)

作為自動化工具, 在此選擇不要輸入密碼形式, 運行後會多了一對兩個檔案(私鑰和公鑰)在用戶帳號下的.ssh/id\_rsa & .ssh/id\_rsa.pub

在git bash環境下看

$ ls ~/.ssh #可以看到兩個檔案:

id\_rsa

id\_rsa.pub

但在windows10可能是這樣的絕對目錄:

/c/Users/louis\_huang/AppData/Roaming/SPB\_Data/.ssh/id\_rsa

/c/Users/louis\_huang/AppData/Roaming/SPB\_Data/.ssh/id\_rsa.pub

兩個帳號產生的ssh keys分別是:

id\_rsa

id\_rsa.personal

id\_rsa.pub

id\_rsa.pub.personal

**※私鑰代理及劃分主機密鑰**

啟用ssh keys代理並增加你要代理登入的服務ssh key:

$ ssh-keygen bash

$ ssh-add ~/.ssh/id\_rsa

$ ssh-add ~/.ssh/id\_rsa.personal

$ ssh-add –l #list有清單表示代理服務keys

讓ssh代理區分密鑰: 建立或增加.ssh/config文檔, 使他的內容如下

$ cat ~/.ssh/config

#預設或私人帳號

Host github.com

HostName github.com

User git

IdentityFile c:/Users/louis\_huang/AppData/Roaming/SPB\_Data/.ssh/id\_rsa.personal

#另一個工作帳號,

Host githubwork

HostName github.com

User git

IdentityFile c:/Users/louis\_huang/AppData/Roaming/SPB\_Data/.ssh/id\_rsa

**※公鑰複製到Github.com->Setting->SSH ans GPG keys**

$ cat ~/.ssh/id\_rsa.pub #將此帳號公鑰內容複製到Github.com->Setting->SSH key(增加)

另一個公鑰內容id\_rsa.pup.personal一樣複製到你私人帳號下的Github.com->Setting->SSH key

**※專案改變倉庫repo的指向(對應keys)**

[你的專案倉庫url一般預設都是指git@github.com](mailto:你的專案倉庫url一般預設都是指git@github.com)對應到密鑰key如果有兩個以上的倉庫&密鑰就透過主機設置名稱區分cat ~/.ssh/config將本地倉庫的密鑰指向不同主機名稱.

進入本地倉庫有個隱藏目錄.git

$ cat .git/config

…

[remote "origin"]

url = git@github.com:repo\_name/project.git

fetch = +refs/heads/\*:refs/remotes/origin/\*

…

根據你host名稱改為githubwork

…

[remote "origin"]

url = githubwork: repo\_name/project.git

fetch = +refs/heads/\*:refs/remotes/origin/\*

…

**※測試倉庫位置**

Git bash到你的空目錄區測試一下, clone下載你工作帳號的倉庫(已存在的repo)

$ git clone git@github.com:repo\_name/project.git

And Or 修改一下README.md內容做更動(若是工作帳號的專案, 則修改專案下的nano .git/config [主機名稱git@github.com](mailto:主機名稱git@github.com) 為githubwork (如前面敘述), 私人帳號專案則用原來的認證主機名稱, 如此區分別要認證的的密鑰

$ git add .

$ git commit –am “update”

$ git push

開啟另一個Git bash到你的空目錄區測試一下, clone下載你私人帳號的倉庫repo

跟前一個動作一樣, 你會發現兩個專案所修改內容已經push到兩個帳號倉庫