窗体顶端



窗体底端

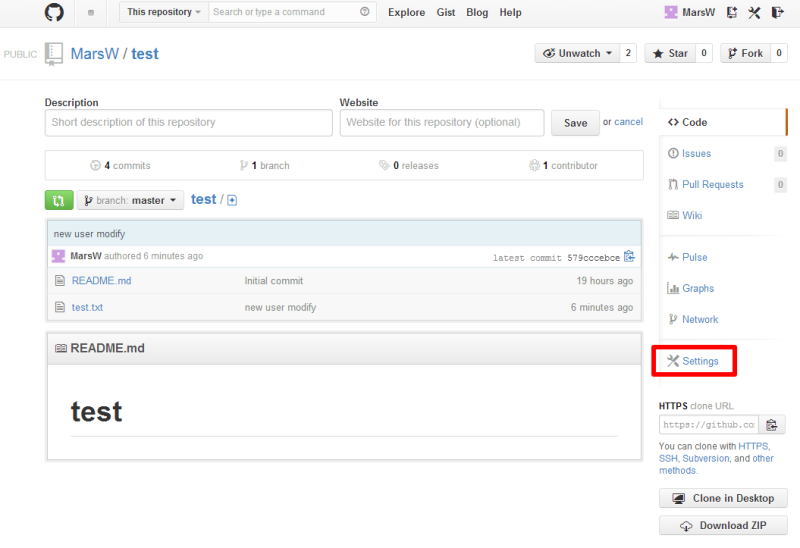
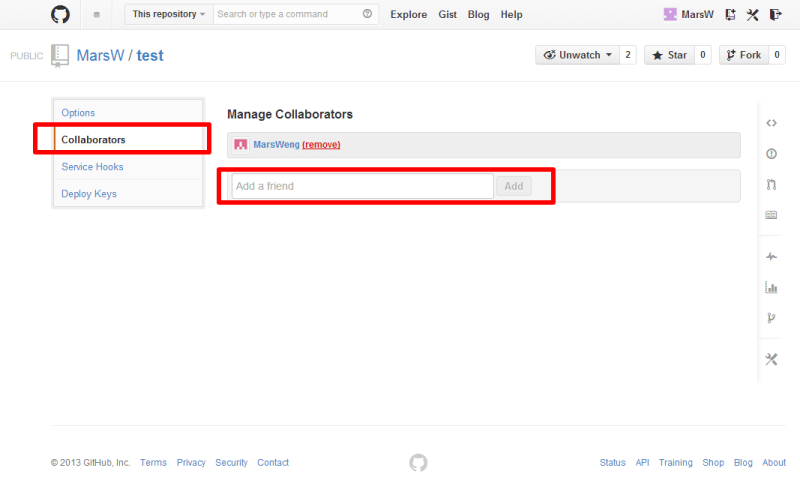
[**I try | MarsW**](http://tech.marsw.tw/)

http://tech.marsw.tw/blog/2013/08/17/git-notes-github-n-person-cooperation-settings

[**Git 筆記 - Github 多人合作(設定)**](http://tech.marsw.tw/blog/2013/08/17/git-notes-github-n-person-cooperation-settings)

**Collaborators**

Collaborators適用於一個team之間的合作。  
負責開這個repository的人，進入settings，加入collaborators的帳號

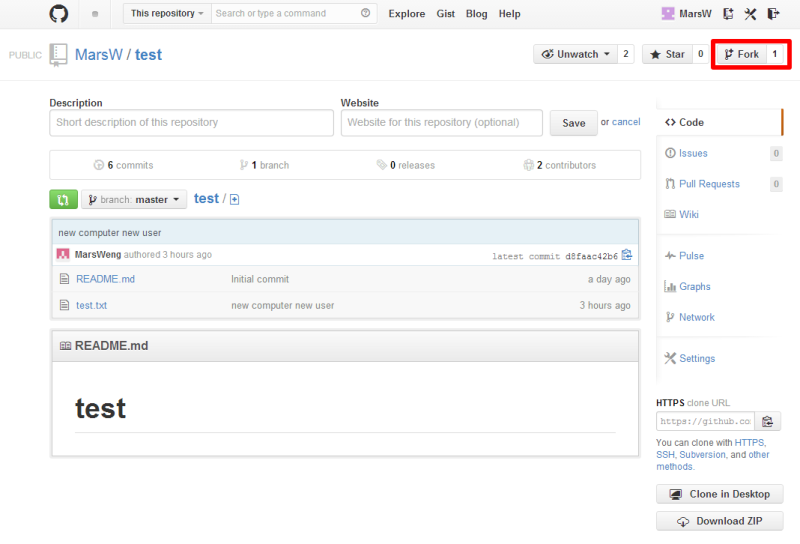
  
  
這樣所有collaborators名單中的人，在git push的時候，可以輸入自己的帳密，而不是擁有repository人的帳密。

**流程會像這樣：**

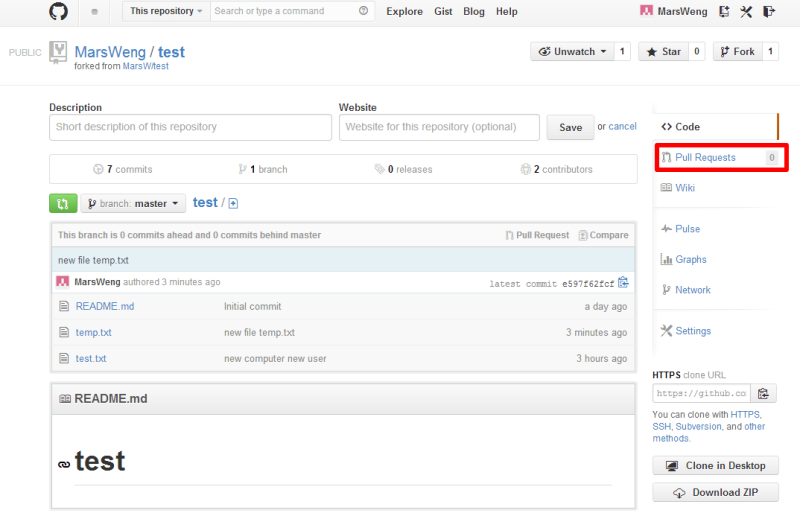
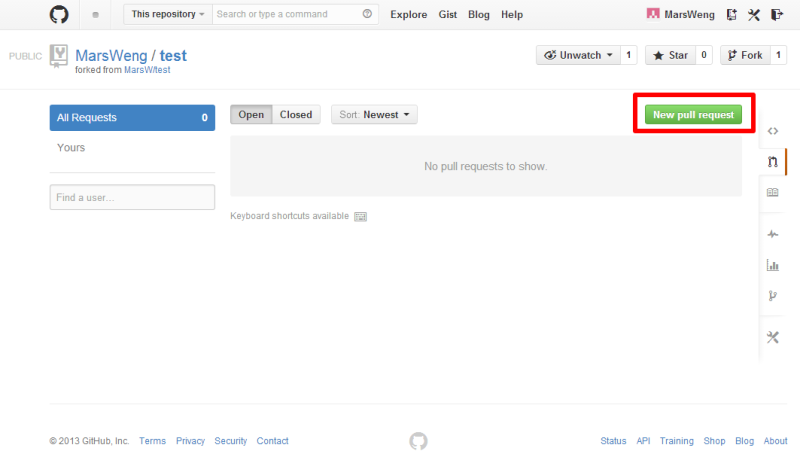
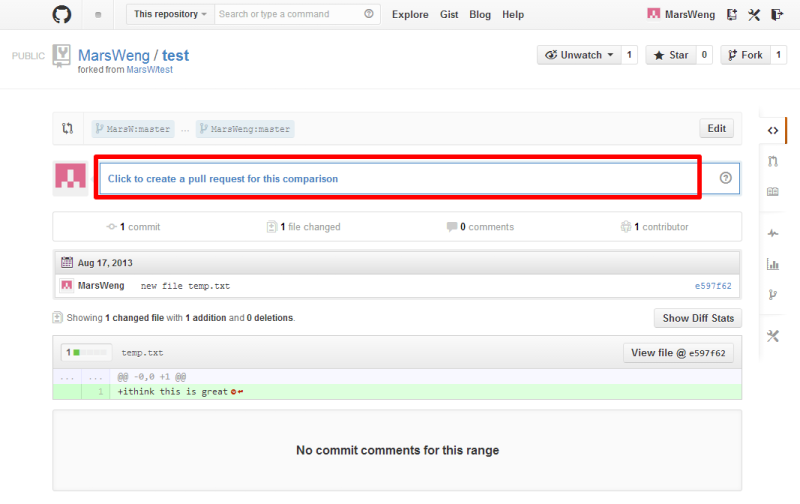
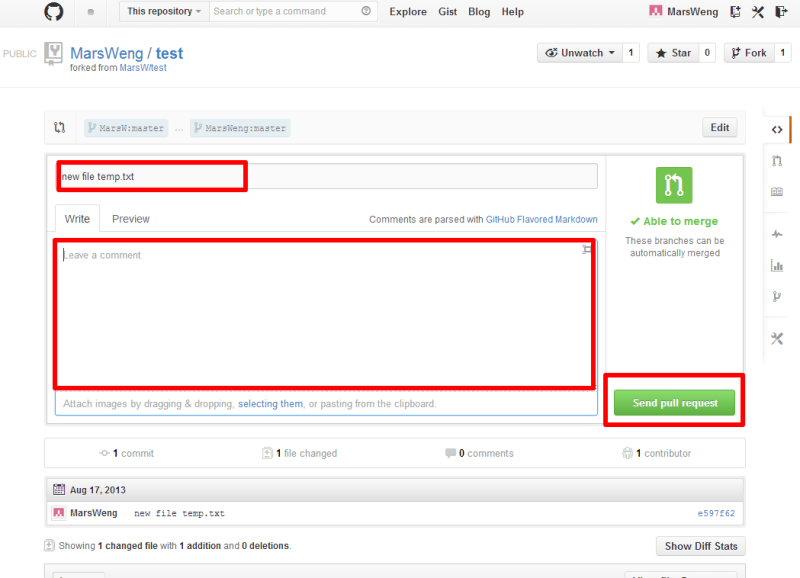
A擁有repository，想要B也能更新自己的repository，就把B加入collaborators，  
這樣B要一起合做這份code的時候，先按照[前一篇](http://tech.marsw.tw/blog/2013/08/16/git-notes-github)的流程設定  
B初始化的時候，\*\*注意clone URL 是 A 的 repo 的 clone URL\*\*，  
然後B就可以在自己的本機盡情修改code (用branch、commit、merge)  
最後在push的時候輸入B自己的帳密，  
會更新A的repository(A仍就可以繼續更新自己的repository)  
**但B自己的github帳戶並不會有一份和A相同的repository(只會有A的repo連結)**

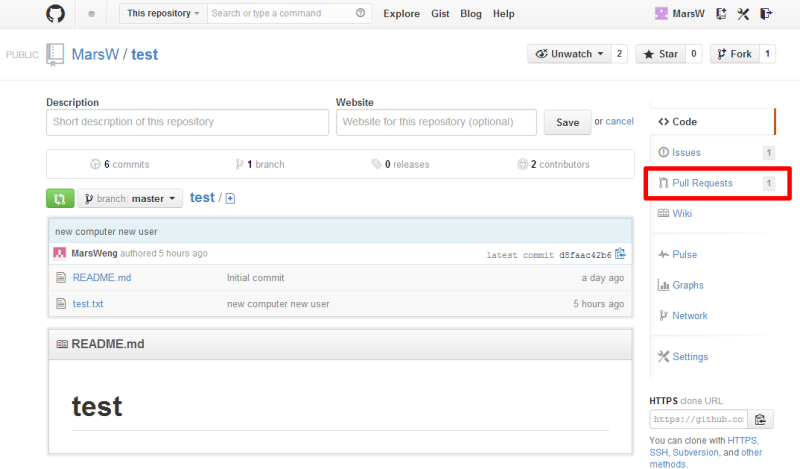
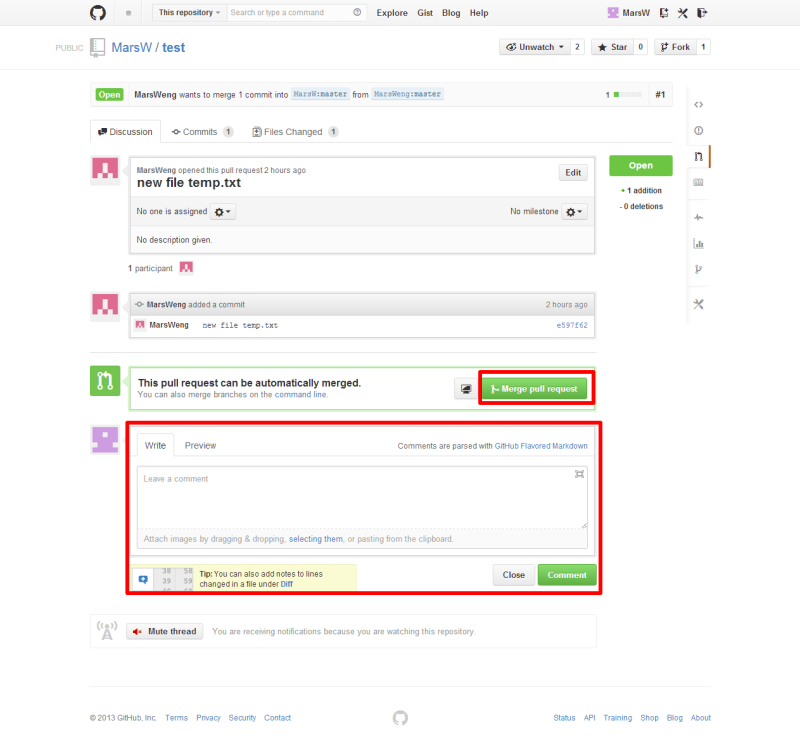
**Fork & Pull request**

另外還有另外一種多人合作方式是 Fork & Pull request  
情境像是 A 擁有一個repo a，B看到覺得還不錯   
所以就fork一個到自己的repo中，這裡叫做b (雖然a,b實際上名字一樣，不過用a,b稱呼比較容易搞懂)

以下圖中 A 對應到 MarsW ； B 對應到 MarsWeng  


Then,就如同以往按照流程，初始clone自己的repo b下來，  
然後B就可以盡情修改code(branch、commit、merge、push)，  
每次 B push更新repo，都只會更新自己的repo b，並不會影響到A的repo a

如果哪天B覺得自己寫的功能很不錯，可以幫助到A，想跟A分享  
那就可以發一個pull request，問問A想不想要這一份  
  
  
  


A 那邊就會收到這則 pull request，  
  
如果覺得ok，用線上merge，就會將repo b合併到repo a上 (Collaborators也有能merge的權限)  
或是覺得有問題，就可以用comment討論  


[← Git 筆記 - Git初始設定 & Github入門](http://tech.marsw.tw/blog/2013/08/16/git-notes-github) [Git 筆記 - 多人合作(流程) →](http://tech.marsw.tw/blog/2013/08/18/git-notes-cooperation-flow)