**GitHub实战系列汇总：**[http://www.cnblogs.com/dunitian/p/5038719.html](http://www.cnblogs.com/dunitian/p/5038719.html" \t "_blank)

**很多人问，明明有git gui 和 github可以直接图形化操作的吗？全部指令干啥？？？**

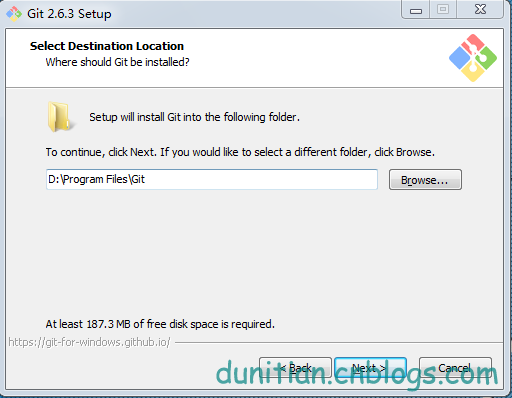
**其实就是为了通用和熟悉git，linux里面照样这样用，多熟悉点基础指令很有用的，**

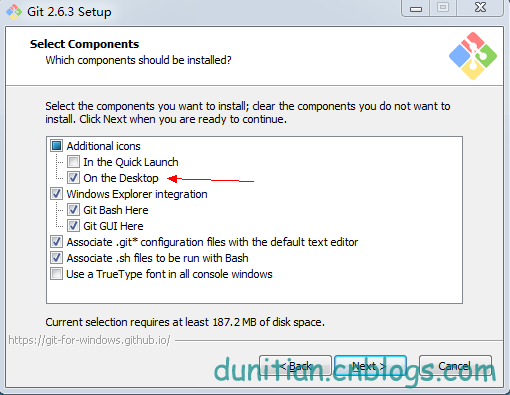
**如果觉得顿时不开心了、无爱了==>推荐你快速入门：**http://www.imooc.com/learn/390

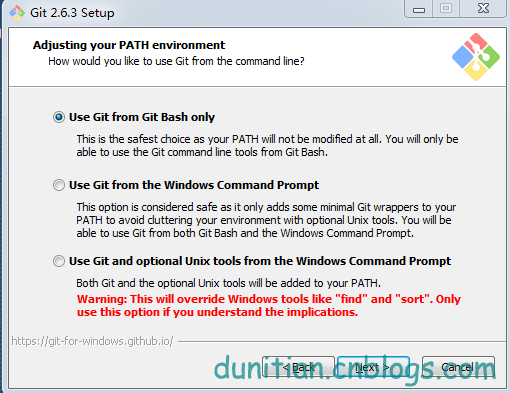
装系列：

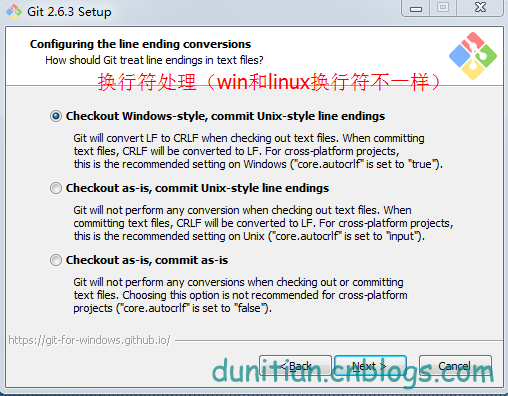
软件下载：http://git-scm.com/download/

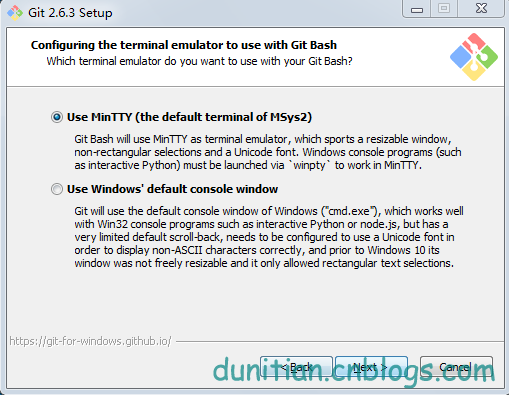
环境搭建：（比较简单，看图）

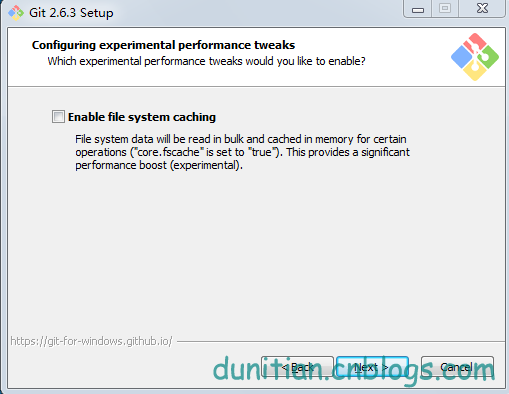


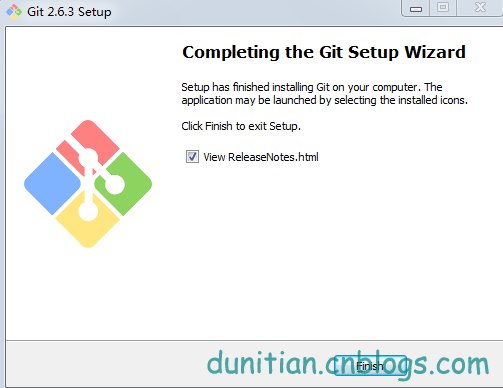


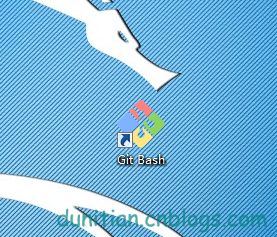






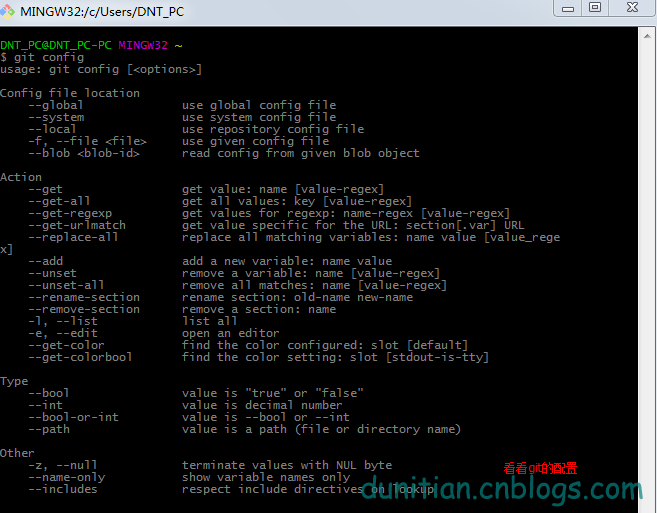






下面是命令模式，需要一点点linux基础（**[Linux基础学习](http://www.cnblogs.com/dunitian/p/4822807.html" \t "_blank)**），没有也没事，看详解

**1.我们看看git的配置都有哪些：**



**git config**  
———————————————————————————————————usage: git config [<options>]

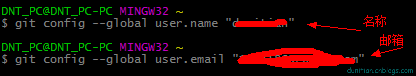
Config file location  
--global use global config file  
--system use system config file  
--local use repository config file  
-f, --file <file> use given config file  
--blob <blob-id> read config from given blob object

Action  
--get get value: name [value-regex]  
--get-all get all values: key [value-regex]  
--get-regexp get values for regexp: name-regex [value-regex]  
--get-urlmatch get value specific for the URL: section[.var] URL  
--replace-all replace all matching variables: name value [value\_rege x]  
--add add a new variable: name value  
--unset remove a variable: name [value-regex]  
--unset-all remove all matches: name [value-regex]  
--rename-section rename section: old-name new-name  
--remove-section remove a section: name  
-l, --list list all  
-e, --edit open an editor  
--get-color find the color configured: slot [default]  
--get-colorbool find the color setting: slot [stdout-is-tty]

Type  
--bool value is "true" or "false"  
--int value is decimal number  
--bool-or-int value is --bool or --int  
--path value is a path (file or directory name)

Other  
-z, --null terminate values with NUL byte  
--name-only show variable names only  
--includes respect include directives on lookup

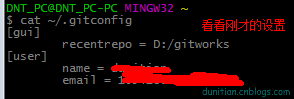
————————————————————————————————————**2.设置名字：**

  
**git config --global user.name "你的名字"**

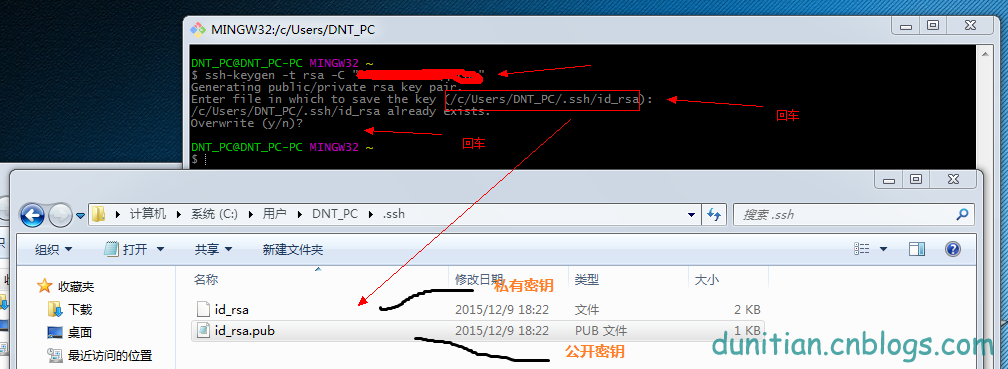
**3.设置邮箱：**  
**git config --global user.email "你的邮箱"**

**4.输出更显目的设置**  
**git config --global color.ui auto**

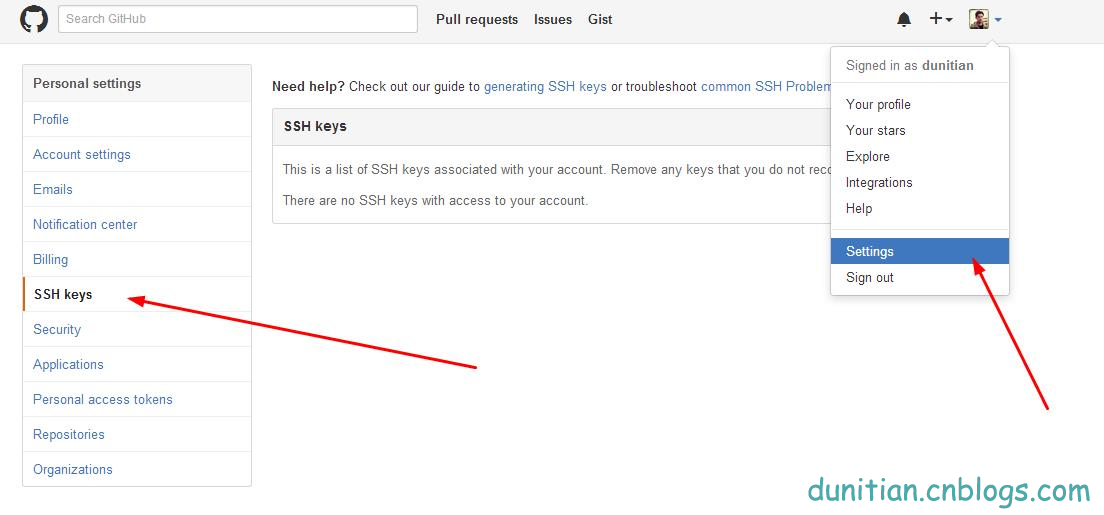
**5.如果忘记自己的配置，就用这个命令来查看一下**

  
**cat ~/.gitconfig**

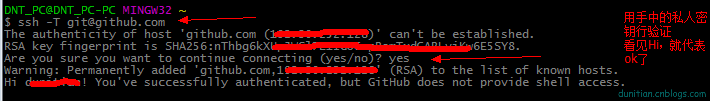
————————————————————————————————————[user]  
name = 你的名字  
email = 你的邮箱  
[color]  
ui = auto  
————————————————————————————————————**6.设置SSH Key**

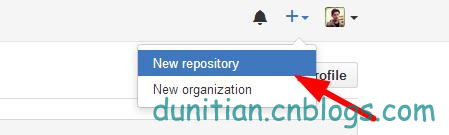
  
**ssh-keygen -t rsa -C "你的邮箱"**  
Generating public/private rsa key pair.  
Enter file in which to save the key (/c/Users/DNT\_PC/.ssh/id\_rsa): 回车一下  
/c/Users/DNT\_PC/.ssh/id\_rsa already exists.  
Overwrite (y/n)? 回车一下  
————————————————————————————————————**7.记不得key用这个指令：**

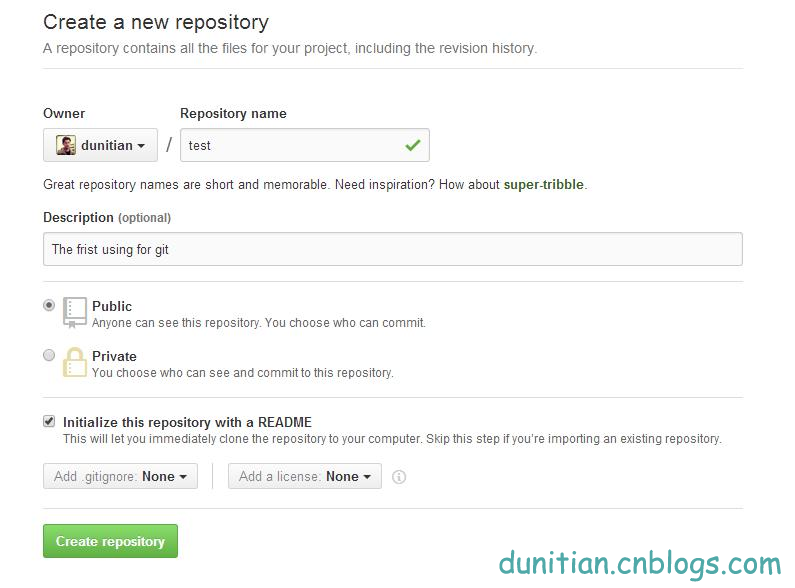
**$ cat ~/.ssh/id\_rsa.pub**  
————————————————————————————————————ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQD3joLDvLaBHnP4aMJwcjo6gjcipBoOnCzETPkQBu+LdBit8L3CSltQ6AhgOL8xcUHxR+ZojdNhd0XXvOfIfuPJVBH57dqptvAeqDkQHiBE5lX2/7pjPVPHSeTiQd0ijYtc1HxtqMo4U++sR6M7QYXPFnHBogUmZdxItVWr\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*4H2h19aIUImZU2KLndgP1AYGFh1FsprWO0oa6ebsIsPGtgrtHqBfHd9e2yF0/1fIFhidXgGvgmt4K9nO0WJ24vW\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
————————————————————————————————————**8.在github里面添加密钥（公共）**

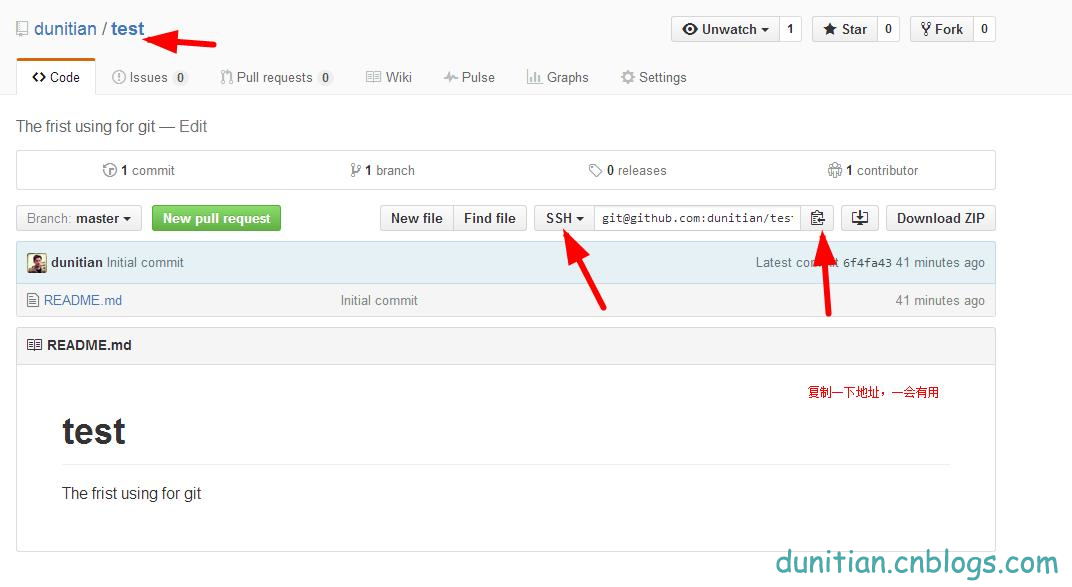


  
————————————————————————————————————**9.用手中的私人密钥和github验证**

  
**ssh -T git@github.com**  
————————————————————————————————————The authenticity of host 'github.com (192.30.252.128)' can't be established.  
RSA key fingerprint is SHA256:nThbg6k\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*viKw6E5\*\*\*\*\*\*\*\*.  
Are you sure you want to continue connecting (yes/no)? 输入yes  
Warning: Permanently added 'github.com,192.30.252.128' (RSA) to the list of known hosts.  
Hi 你的姓名! You've successfully authenticated, but GitHub does not provide shell access.  
————————————————————————————————————**10.在github里面创建一个公开仓库（私有的收费）并初始化仓库（最下面的复选框，最下面的两个下拉列表后面说）**

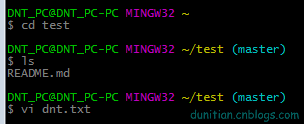


  
————————————————————————————————————**11.复制一份github ssh库的地址，一会儿有用**

  
————————————————————————————————————**12.克隆一份到本地**

**git clone git@github.com:dunitian/test.git（刚才的地址）**

————————————————————————————————————Cloning into 'test'...  
Warning: Permanently added the RSA host key for IP address '192.30.252.131' to the list of known hosts.  
remote: Counting objects: 3, done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (3/3), done.  
Checking connectivity... done.  
————————————————————————————————————**13.写点东西测试一下**

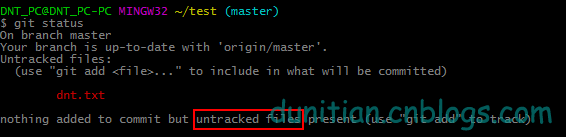
  
**cd test**（项目名，注意大小写）

DNT\_PC@DNT\_PC-PC MINGW32 ~/test (master)  
**ls**（查看当前目录有哪些东西）  
README.md

DNT\_PC@DNT\_PC-PC MINGW32 ~/test (master)  
**vi dnt.txt** （如果有dnt.txt的文件就打开，没有就创建一个dnt.txt的文件）



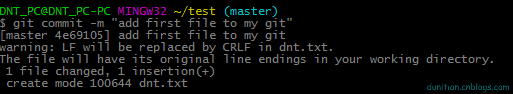
输入你想写的东西，i 进入编辑模式 ，按esc 输入 :wq 并回车 是保存   
————————————————————————————————————**14.查看git的状态 （没提交，所以是untracked files状态）**

  
**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)

dnt.txt

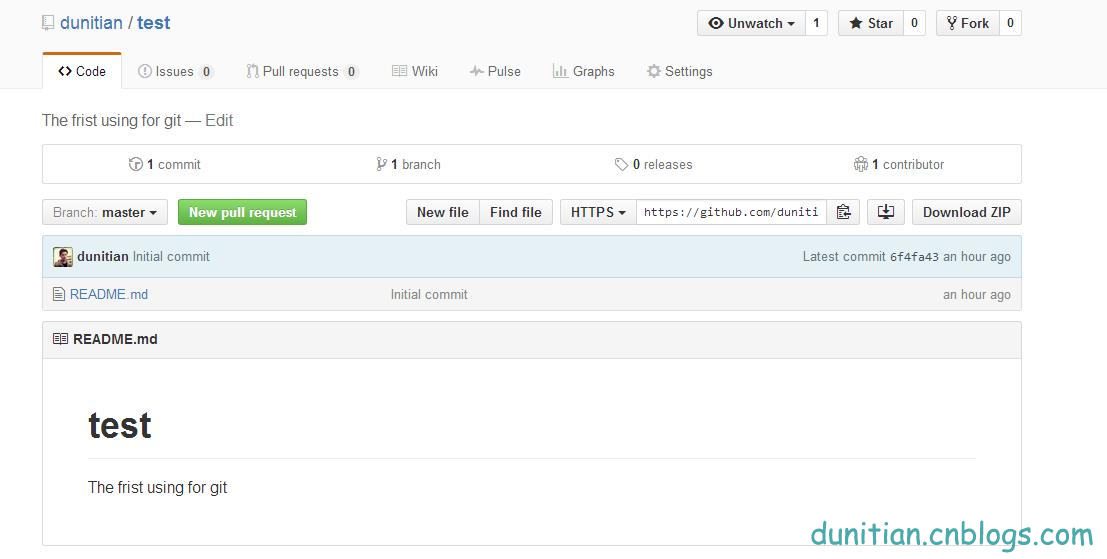
nothing added to commit but untracked files present (use "git add" to track)  
————————————————————————————————————**15.把文件提交到仓库**

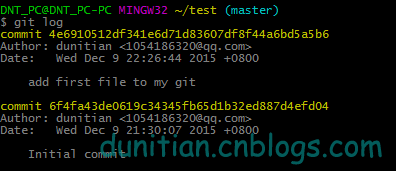
https://images2015.cnblogs.com/blog/658978/201512/658978-20151209230019090-1589667690.png  
**git add dnt.txt （添加到暂存区）**

****

**git commit -m "add first file to my git" (提交;引号里面是注释)**  
————————————————————————————————————[master 4e69105] add first file to my git  
warning: LF will be replaced by CRLF in dnt.txt.  
The file will have its original line endings in your working directory.  
1 file changed, 1 insertion(+)  
create mode 100644 dnt.txt  
———————————————————————————————————

**16.到github看看，发现这个时候是木有添加我们的文件的(不要怕)**

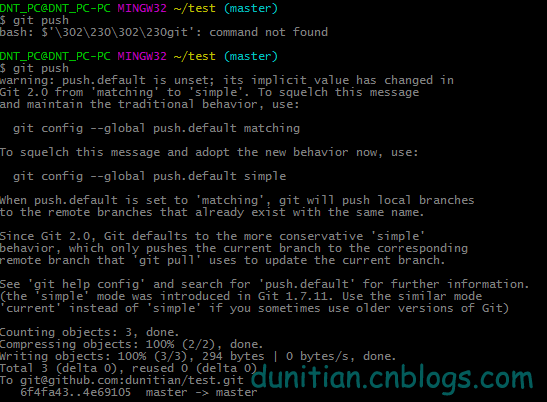
  
————————————————————————————————————**17.查看提交日记 （吓死宝宝了，赶紧看看log）**

  
**git log**  
————————————————————————————————————commit 4e6910512df341e6d71d83607df8f44a6bd5a5b6  
Author: dunitian <1054186320@qq.com>  
Date: Wed Dec 9 22:26:44 2015 +0800

add first file to my git

commit 6f4fa43de0619c34345fb65d1b32ed887d4efd04  
Author: dunitian <1054186320@qq.com>  
Date: Wed Dec 9 21:30:07 2015 +0800

Initial commit  
————————————————————————————————————**18.原来是要push一下，更新github（如果发现command not found之类的问题再输入一遍就ok了）**

  
**git push**  
————————————————————————————————————————  
warning: push.default is unset; its implicit value has changed in  
Git 2.0 from 'matching' to 'simple'. To squelch this message  
and maintain the traditional behavior, use:

git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

git config --global push.default simple

When push.default is set to 'matching', git will push local branches  
to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'  
behavior, which only pushes the current branch to the corresponding  
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.  
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode  
'current' instead of 'simple' if you sometimes use older versions of Git)

Counting objects: 3, done.  
Compressing objects: 100% (2/2), done.  
Writing objects: 100% (3/3), 294 bytes | 0 bytes/s, done.  
Total 3 (delta 0), reused 0 (delta 0)  
To git@github.com:dunitian/test.git  
6f4fa43..4e69105 master -> master  
————————————————————————————————————————