Eclipse git插件的使用

2016年9月 07

9:55

1.生成ssh key

计算机生成了可选文字:
eferences 
type filter text 
SSH2 
N otificatio ns 
Pe r s pectives 
Sea rch 
Secu rity 
Service Policies 
Startup and Shutd 
Tracing 
UI Responsivenes± 
Web Browser 
Workspace 
Code Recommenders 
Data Management 
Help 
Install 'Update 
Java 
Java EE 
Java Persistence 
JavaScript 
- Iukai001 
SSH2 
Configuration options for SSH2: 
Key Management Known Hosts Authentication Methods 
Gener 
fene rate DSA Keym Gene ra te RSA Keym Load Existing Key„j 
You can paste this public key into the remote authorized keys file: 
ssh-rsa AAAA83NzaC1yc2EAAAADAQA8AAAAgQCvUJrskk/6+ 
Key Exchange Methods 
+0U6Hb1mWauvfGVHhOwWWJhcIfK8PhWß51Ak36R72iDx/dJKyzIcWYKN3YrOSuAb8uXSPPpIrrXI 
Fingerprint: 
Comment: 
Pass phrase: 
Confirm passphrase: 
1:83: I:fb 
RSA-1024 
Fport Via SFI 
Restc 
Ca ncel 

生成之后 apply然后save

2.在github上登记ssh key

计算机生成了可选文字:
Di sconf 
e 10.1.1.83 
/praflle/keys 
GitLab 
Back to Dashboard 
Profile 
Account 
Applications 
Password 
Notifications 
SSH Keys 
Preferences 
Audit Log 
User Iukai 
activitv 
Disconf - 
.3f:gø 
5a:9c• 
.47: ad. 
c7:11: 
ce:84 
X 
Di sconf 
SSH Keys Sett... 
Dubbo Admin 
Di sconf 
Di sconf 
Di sconf 
Search 
Di sconf 
Settings 
SSH Keys 
Before you can add an SSH key you need to generate it. 
O 
e 
Title 
Iukai@lukai 
test 
Fingerprint 
.dd. 
c4• 
•25: 
64 • 
.eb. 
•16: 
Added at 
added 2 months ago 
added 4 months ago 
'm.aum.authorize.allowNul •A v *11m , 
Add SSH Key 
Remove 
Remove 
9:53 
2016/9/7 

3.导入git中工程

本地git的搭建

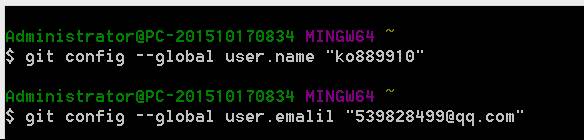
2016年9月 07

14:26

1.下载安装MsysGit\_70752

2.打开gitbash命令行

3.输入用户名和邮箱,作为一个标识(作为git分布式系统的一个单元)



注意:git config --global 参数,表示这台机器上有所得git仓库都会使用这个配置,当然你也

可以令某个仓库对应不同的用户名和邮箱

4创建版本库(即仓库)

输入指令git init就会将当前目录变为git仓库,会多一个.git隐藏文件夹,用来控制git的管理

5.添加文件到版本库中

分为两步

1.git add +文件名(将文件放入本地缓存区)

2.git commit -m +"XXX"+文件名(从缓存区放入版本库中)

-m表示添加提交时候的说明 XXX就是说明



为什么分成两步?

因为可以多次add 一次commit一次提交

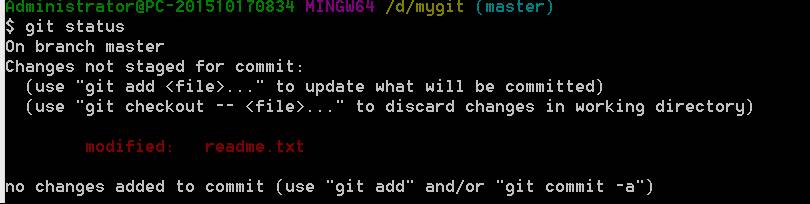
$ git add file1.txt

$ git add file2.txt file3.txt

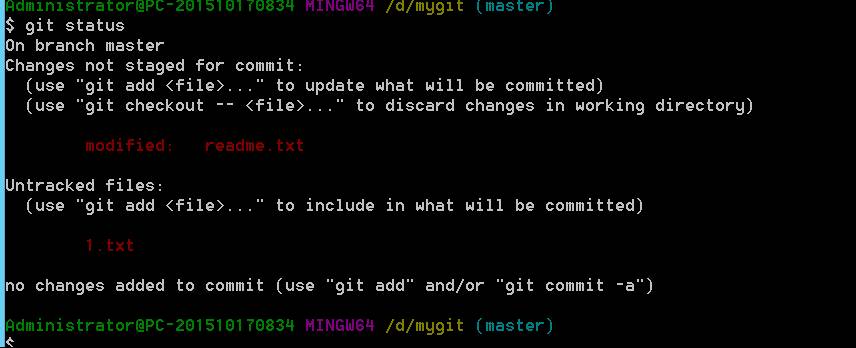
$ git commit -m "add 3 files."

6.查看本地与版本库的区别

Git status (比较的是本地和版本库中的区别,或者本地缓冲区与版本库的区别)



上图显示readme.txt有改动

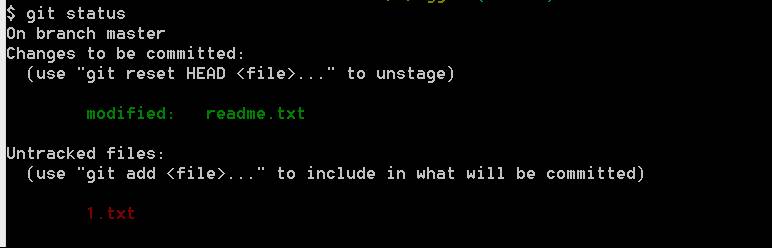


上图显示readme.txt有改动

新增了1.txt

而具体的文件差异可以通过  
如果将readme.txt

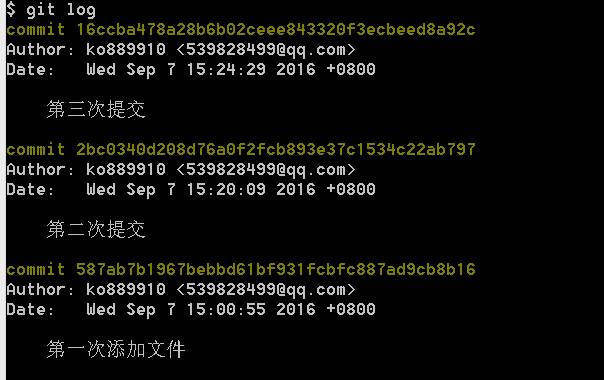
再次来运行git status命令发现



由红色变为了蓝色,代表差异已经在缓存空间中了

7.查看版本记录

Git log



黄色部分是uuuid,是每一次操作产生的

如果在后面添加–-pretty=oneline会忽略提交者和日期



8.回退版本

方法1.git reset -hard HEAD^(会退到上一个版本)

git reset -hard HEAD^^(会退到上2个版本)



方法2.git reset -hard HEAD~100(会退到上100个版本)

9.回退版本之后又撤销回退,重新回到最新的版本

1.首先根据 git reflog查看所有的版本日志

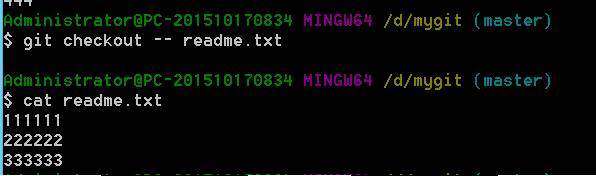
根据上图可以发现目前head指在了最新版本号的前一个版本-2bc0340上,所以得知最新版本是16ccba4

2.然后使用git reset --hard +版本号回到最新版本



10.撤销修改保持和版本库中一致

Git checkout -- +文件名(只适用于还没add到缓冲区的修改,如果不加-- 就是变成增加分支的命令了)



11.删除文件

直接使用rm 文件名

然后使用add 和commit命令提交就行

1.下载安装MsysGit\_70752

2.打开gitbash命令行

3.输入用户名和邮箱,作为一个标识(作为git分布式系统的一个单元)

计算机生成了可选文字:
Administrator@PC-2015101 70834 
$ git config 
-global user . name "k088991@" 
Administrator@PC-2015101 70834 
$ git config 
--global user . emalil "539828499@qq.com•• 

注意:git config --global 参数,表示这台机器上有所得git仓库都会使用这个配置,当然你也

可以令某个仓库对应不同的用户名和邮箱

4创建版本库(即仓库)

输入指令git init就会将当前目录变为git仓库,会多一个.git隐藏文件夹,用来控制git的管理

5.添加文件到版本库中

分为两步

1.git add +文件名(将文件放入本地缓存区)

2.git commit -m +"XXX"+文件名(从缓存区放入版本库中)

-m表示添加提交时候的说明 XXX就是说明

计算机生成了可选文字:
Administrator@PC-2015101 70834 
/d/mygit (master) 
$ git commit -m " % — readme. txt 
[master (root-commit) 587ab7b] 
1 file changed, 1 insertion(+) 
create mode readme. txt 

为什么分成两步?

因为可以多次add 一次commit一次提交

$ git add file1.txt

$ git add file2.txt file3.txt

$ git commit -m "add 3 files."

6.查看本地与版本库的区别

Git status

计算机生成了可选文字:
d mygit 
$ git status 
On branch master 
Changes not staged for commit: 
(master) 
(use "git add < file). . . " 
(use "git checkout - 
to update what will be committed) 
to discard changes in working directory) 
modified: 
readme. txt 
no changes added to commit 
(use 
"git add" 
and/or 
•git commit 
-a") 

上图显示readme.txt有改动

计算机生成了可选文字:
d myglt 
$ git status 
On branch master 
Changes not staged for commit: 
(master) 
(use "git add <file>.. " 
(use "git checkout - 
modified: 
Untracked files: 
readme. txt 
to update what will be committed) 
to discard changes in working directory) 
to include in what will be committed) 
(use "git add < file). . . " 
. txt 
no changes added to commit 
(use "git add" and/or "git commit 
-a") 
Administrator@PC-2015101 70834 
/d/mygit (master) 

上图显示readme.txt有改动

新增了1.txt

而具体的文件差异可以通过  
如果将readme.txt

再次来运行git status命令发现

计算机生成了可选文字:
$ git status 
On branch master 
Changes to be committed: 
(use "git reset HEAD < file). . . " 
to unstage) 
modified: 
Untracked files: 
readme. txt 
(use "git add < file). . . " 
. txt 
to include in what will be committed) 

由红色变为了蓝色,代表差异已经在缓存空间中了

9.回退版本之后又撤销回退,重新回到最新的版本

1.首先根据 git reflog查看所有的版本日志

计算机生成了可选文字:
Administrator@PC 
$ git reflog 
2bc0340 
2bc0340 
587ab7b 
/d/mygit (master) 
-2015101 70834 
reset: 
commit: 
commit: 
commit 
moving to HEADA 
(initial): 

根据上图可以发现目前head指在了最新版本号的前一个版本-2bc0340上,所以得知最新版本是16ccba4

2.然后使用git reset --hard +版本号回到最新版本

10.撤销修改保持和版本库中一致

Git checkout -- +文件名(只适用于还没add到缓冲区的修改,如果不加-- 就是变成增加分支的命令了)

计算机生成了可选文字:
Administrator@PC- 
$ git checkout - 
Administrator@PC- 
$ cat readme. txt 
222222 
333333 
2015101 70834 
readme. txt 
2015101 70834 
/d/mygit 
MINGW64 
/d/mygit 
MINGW64 
(master) 
(master) 

Git log

计算机生成了可选文字:
commit 16ccba478a28b6b02ceee843320f3ecbeed8a92c 
Author: 1<0889910 
Wed sep 7 2016 +0800 
commit 2bc0340d208d76aOf2fcb893e37c1534c22ab797 
Author: 1<0889910 
Wed sep 7 2016 +0800 
commit 587ab7b1967bebbd61bf931fcbfc887ad9cb8b16 
Author : 
1<0889910 
Wed sep 7 2016 +0800 

黄色部分是uuuid,是每一次操作产生的

如果在后面添加–-pretty=oneline会忽略提交者和日期

计算机生成了可选文字:
•t 1 g - -pretty:oneline 
1 6ccba478a28b6b02ceee843320f3ecbeed8a92c 
2bc0340d208d76aOf2fcb893e37c1534c22ab 797 
fcbfc887ad9cb8b16 
Administrator@PC-2015101 70834 
/d/mygit (master) 

远程仓库

2016年9月 07

16:19

1.github的初步配置,详情参照github

2.让本地仓库与github上的库进行同步

(达到共享库,协同开发的目的)

a.登陆github并创建库

计算机生成了可选文字:
Pull requests 
SSH keys 
G [St 
New repository 
Import repository 
New Organization 
This is list 耐 SSH keys associated with your account. Remove any keys that you 
q•ux•ang 
Fingerprint ： fe ： 5 ： 71 ： 3 b ： 5 5 ： 39 ： 55 ： 55 ： 2 ： 9 8 ： IC ： 1 巴 ： 8 ： 51 ： 8 巴 ： 6f 
Added o n 7 Sep 201 6 _ Never used 
@ Check out o u r guide to generating SSH keys or troubleshoot common SSH Problems. 
GPG keys 
There ar 巳 no GPG keys With access to your account. 
@ Learn how to generate GPG key and add ittoyour account. 
New GPG key 

b.

现在，我们根据GitHub的提示，在本地的testgit仓库下运行命令：

计算机生成了可选文字:
0 
This rep 0 Search 
k0889910 /jllw 
Pull requests 0 
目 《 Wiki 
Pull requests 
Pulse 
G [St 
Graphs 
0 Unwatch 
0 Settings 
*Star 
0 
〈 > Code 
． Issues 0 
Quick setu p 一 if you've done this kind of thing before 
〔 ± 〕 SetupinDesktop Or HITPS SSH https 匚 0m ／ k0889g12 ／ it 
We recommend every repository include 3 README, LICENSE, and .gitignore. 
灬 Or create a new repository On the command line 
echo “ # j11 “ “ > > README md 
81t 1 n It 
git add README md 
first commit 
81t commit 一 m 
git r emot e add origin https://github. com/k0889918/j 11 ． git 
git push 一 u origin master 
… or push an existing repository from the command line 
git r emot e add origin https://github. com/k0889918/j 11 ． git 
git push 一 u origin ma st e r 
… or import code from another repository 
You can initialize this repository with code from 3 Subversion, Mercurial, or TFS prp 」 ect. 
Importcode 
0 
ProTip! use the LIRI- for this page when adding GitHub as 3 remote. 

在本地仓库目录下运行命令:

git remote add origin <https://github.com/ko889910/jllw.git>

然后调用

Git push -u origin master将master分支放在远程仓库中(

-u参数的解释  
因为你本地的分支有可能可以关联多个远程分支,如果这样每次都要选择关联的

远程分支,加了-u就是选择了默认的关联分支,以后提交下载跟新就不需要指定了

另外以上两个命令加起来等效于以下命令:

git push <https://github.com/ko889910/jllw.git> master ( git push <远程主机名> <本地分支名>:<远程分支名>)

本地分支名不写默认就是master

Git remote add origin +地址

的语句的含义是给origin赋值,即origin=地址,方便后面的调用(是全局变量,以后都可以使用)

)

计算机生成了可选文字:
$ d r 
$36@RTLOG$ 
O nSSH 
Username for 'https://github.com/ 
d m 1 n 1 s t r a t 0 r 但 P C ． 201 51 @1 7e83 
$ git remote add origin https://91thub.com/k088991@/〕 11国．g1t 
36e 安 全 浏 览 器 下 载 KwDown10adx 
bbtT001BackUp 
DrluerBackUp 
． 2@1 51 @1 7e83 巪 
． 2@1 51 @1 7e83 巪 
C ． 201 51 @1 7e83 巪 
m u n 
myglt 
MINGW64 
OC 巪 〕 
Program\ Files\ 〔 × 86 〕 
System\ Uolume\ Informatz 
dd 
0 
illw.git 
Cancel 
$ d r 
/d/mygit 〔 master 〕 
MINGW64 
0 r 1 g 1 n https://91thub.com/k088991@/〕 11国．g1t 
/d/mygit 〔 master 〕 
MINGW64 
/d/mygit 〔 master 〕 
MINGW64 
1 ． t × t readme. t × t 
d m 1 n 1 s t r a t 0 r P C ． 2@1 51 @1 7e83 巪 
f a t a 1 ： remote 0 r 1 g 1 n already e × 1 s t s ． 
/d/mygit 〔 master 〕 
d m 1 n 1 s t r a t 0 r 但 P C ． 201 51 @1 7e83 
$ git push ． u origin master 

计算机生成了可选文字:
/d/mygit 〔 master 〕 
A d m 1 n 1 s t r a t 0 r 但 P C ． 201 51 @1 7e83 巪 
$ git push ． u origin master 
CountIng 0b 〕 ects: 1 3 ， done. 
D e 1 t a compression using up t 0 8 threads. 
Compressing objects: 1@@% 〔 5 / 5 〕 ， done. 
Writing objects: 1@@% 〔 1 3 / 1 3 〕 ， 956 bytes 《 @ bytes/s, 
T 0 t a 1 1 3 〔 delta @〕 ， reused @ 〔 delta @〕 
TO https://91thub.com/k088991@/〕 11国．g1t 
done 。 
[new branch] 
master ． 〉 master 
Branch master set up t 0 track remote branch 
master from 0 r 1 g 1 n ． 
/d/muQIt 〔 m a S t e r 〕 
A d m 1 n 1 s t r a t 0 r 但 P C ． 201 51 @1 7e83 巪 

计算机生成了可选文字:
李 上 路 
用 址 fi ／ ／ / C sers / k 。 88 一 爰 爰 
0 
This rep 0 Search 
k0889910 /jllw 
queue ： //F00吧众R 
queue ： //FIrstQueue 
Pull requests Issues 
queue://F1rstQueue 
queue ： //FOO.BAR 
GLSt 
0 Unwatch 
*Star 
0 
Fork 
0 
〈 > Code 
． Issues 0 
Pull requests 0 
目 《 Wiki 
1 branch 
Pulse 
堤 交 1 .txt 
堤 交 1 .txt 
Graphs 
0 Settings 
NO d cr 五 0 月 or website provided. - 一 Edit 
4 commits 
0 0 releases 
8 n 〔 h ： master 
New pull request 
k088gg10 交 1 .txt 
固 l.txt 
固 readme.txt 
里 readme.txt 
111111 
222222 
333333 
3334 
惠 0 contributors 
〔 n 已 or dcnvnload - 
Latest 〔 omm 57 已 d982 1 g hours 3g0 
1 9 hours ago 
1 9 hours ago 

3.本地提交到远程库

首先和以前一样使用git add和git commit完成本地库的提交

然后使用git push origin master 完成远程库的提交

计算机生成了可选文字:
A d m 1 n 1 s t r a t 0 r P C ． 2@1 51 @1 7e83 巪 
$ git push origin master 
CountIng 0b 〕 ects: 3 ， done. 
D e 1 t a compression using up t 0 
/d/mygit 〔 master 〕 
MINGW64 
8 threads. 
Compressing objects: 1@@% 〔 2 / 2 〕 ， done. 
Writing objects: 1@@% 〔 3 / 3 〕 ， 31 7 bytes 《 
T 0 t a 1 3 〔 delta @〕 ， reused @ 〔 delta @〕 
TO https://91thub.com/k088991@/〕 11国．g1t 
67ed98e ． ． 21 38 c@7 master ． 〉 master 
@ bytes/s, 
done 。 
/d/mygit 〔 master 〕 
A d m 1 n 1 s t r a t 0 r 但 P C ． 201 51 @1 7e83 巪 

Github

2016年9月 07

16:21

1.什么是github?

一个外网的git管理web网站,可以简化git操作,以及实现外网的git管理

2.如何使用

a.生成sshkey(本地电脑和github通过ssh协议通信)

创建SSH Key。在用户主目录下，看看有没有.ssh目录，如果有，再看看这个目录下有没有id\_rsa和id\_rsa.pub这两个文件，如果有的话，直接跳过此如下命令，如果没有的话，打开命令行，输入如下命令：

计算机生成了可选文字:
/d/mygit 〔 master 〕 
《 d m 1 n 1 s t r a t 0 r 但 P C ． 201 51 @1 7e83 巪 
$ ssh-keygen ． t r S a ． C “ 539828 巪 99 但 qq ． com “ 
Generating publlc/prluate r S a key p a 1 r ． 
Enter f 1 e n which t 0 S a u e the key 〔 /c/Users/Administrator/ .ssh/id_rsa 〕 ： 
Enter passphrase 〔 empty for n 0 passphrase 〕 ： 
Enter S a m e passphrase a g a 1 n ： 
Your 1 d e n t 1 f 1 c a t 10 n has been saved 1 n /c/Users/Administrator/ .ssh/ld_rsa. 
Your public key has been saved 1 n /c/Users/Administrator/ .ssh/ld_rsa.pub. 
The key fingerprint 1 S ： 
SHA256 :n+Kx5cf2ueUpqEAcUGw2tC1 eNJQsuX3QFmDns+0YEtQ 539828 99 qq 。 C 0 m 
The key's randomart Image 1 S ： 
-[RSA 20 巪 8 ] ． 
0 + 0 + 0E00 
-[SHA256]- 

计算机生成了可选文字:
， 这 台 思 
本 證 〔 0 〕 
名 称 
记 rsa 
Users 
兰 前 视 
修 改 凸 期 
2016 ． g ． 7 1034 
2016 ． g ． 7 1034 
2015 ． 12 ． 31 1 5 巧 0 
显 亍 / 过 
2 KB 
Microsoft Offi 〔 已 
known hosts 

说明:

Id\_rsa是私钥用于github验证权限

整个流程如下:  
首先客户端请求登录github,然后github根据你的用户名或者邮箱找到对应的公钥,然后用公钥加密一段信息,并发送给客户端,客户端使用私钥解密之后返回给服务器,服务器确认无误就与客户端建立ssh连接.

b.打开github

进入setting页面

计算机生成了可选文字:
新 李 上 路 
用 址 fi ／ ／ / C sers / k 。 88 一 
0 
Search GitHLlb 
0 2D16 G H 凵 貊 0 
爰 爰 
queue ： //F00吧众R 
queue ： //FIrstQueue 
Pull requests Issues 
queue://F1rstQueue queue ： //FOO.BAR 
GLSt 
Lea rn Git and GitHub without a ny code! 
Using the Hello World guide, you'll create a repository, start a branch, 
write comments, and open a pull request. 
Read the guide 
Start a proJect 
Signed in as k0889910 
Your profile 
Your stars 
Explore 
Integrations 
Settings 
Sign out 
Discover i nteresting projects and 
people to popu late your personal 
news feed. 
Your news feed helps you keep up with recent activity on 
repositories you watch and p 巳 op you fo 日 0 “ 
（ 《 。 》 ） Updates to our Privacy Statement 
We have updated 0 凵 r Privacy Statement! 
Check 0 凵 《 it's n ot boring! 
Your repositories 
F i n cl 已 
监 日 P 凵 bli 匚 
Iktest 
Private 
View 2 new broadcasts 
New repository 
Forks 
Explore GitHub 
Privacy Security Status Help 
0 
disconf-demos-java 
Contact GitHub 厶 剛 Training Shop 引 og About 

进入ssh配置页面

计算机生成了可选文字:
用 址 fi ／ ／ / C sers / k 。 88 一 
0 
Search GitHLlb 
Personal settings 
Profile 
Account 
Emails 
Notifications 
SSH and GPG keys 
Security 
Blocked users 
OAuth applications 
personal tokens 
Repositories 
Organizations 
v replies 
爰 爰 
queue ： //F00吧众R 
queue ： //FIrstQueue 
Pull requests Issues 
queue://F1rstQueue 
G [St 
queue ： //FOO.BAR 
SSH keys 
There ar 巳 no SSH keys with 3 〔 c 巳 “ to your account. 
New SSH key 
· ， 88 0 
Title 
sh 2 一 nis 521 
Add SSH key 
。 'ssh 一 E02551 《 匚 ds 彐 一 sh 彐 2 一 nistp2S 三 ， 'ecdsa-sha2-nistp384', 0 r 匚 ds 彐 一 
@ Check out o u r guide to generating SSH keys or troubleshoot common SSH Problems. 
GPG keys 
There ar 巳 no GPG keys With access to your account. 
@ Learn how to generate GPG key and add ittoyour account. 
New GPG key 

到此完成了github的初步配置