|  |
| --- |
|  |
| **Git 使用手册（开发人员）V1.2** |
|  |
|  |
| 总主编：米新江 |
|  |
| 参加编撰人员：  琚建飞 、张婷 |
|  |
| **2017 年 6 月 18 日 始建** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 修改时间 | 修 改 人(编撰人） | 审 核 人 | 修改内容（编撰内容） |
| V1.0 | 2017-06-18 | 琚建飞 | 米老师、刘文彬 | 编写文档 |
| V1.1 | 2017-08-12 | 张婷 | 米老师、刘文彬 | 添加常用开发操作 |
| V1.2 | 2017-09-21 | 张婷 | 米老师、刘文彬 | 添加简易教程和命令清单 |
| V1.3 | 2018-10-26 | 王雪芬 |  | 修改文档部分操作 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[使用前提 4](#_Toc528317686)

[1下载，安装，配置 4](#_Toc528317687)

[2.生成自己的公钥和私钥 5](#_Toc528317688)

[3.克隆代码 9](#_Toc528317689)

[4.创建工作分支 10](#_Toc528317690)

[5.开始进行功能开发，提交代码 13](#_Toc528317691)

[6.常用命令 16](#_Toc528317692)

[7.Git 简易教程 17](#_Toc528317693)

[8.Git命令清单 18](#_Toc528317694)

# 使用前提

可能有的命令在粘贴的时候英文字符变成了中文字符，所以大家在粘贴命令的时候要注意一下。

# 1下载，安装，配置

Git支持各大平台，如Windows、Linux、Solaris、Mac OS。根据自己的电脑系统下载相应的版本，如果是Windows操作系统，则点击Windows即可。各平台下载链接：<http://git-scm.com/downloads>

双击exe文件进行安装，一直下一步即可完成安装。完成后，会出现三个图标：Git Bash、Git GUI、Git CMD。如下图所示：



安装完成后，我们需要进行两项配置，一项是配置自己的名字，一项配置自己的邮箱地址，配置的目的是为了可以方便查看是谁修改的代码，以做确定身份之用。在桌面任意处右击—选Git Bash Here，在弹出的Git Bash命令框中输入下面的命令：

$ git config --global user.name “Your name” //配置用户名

$ git config --global user.email “Your Email address” //配置用户邮箱地址

例如：

$ git config --global user.name “Micheal”

$ git config --global user.email “15732625435@163.com”

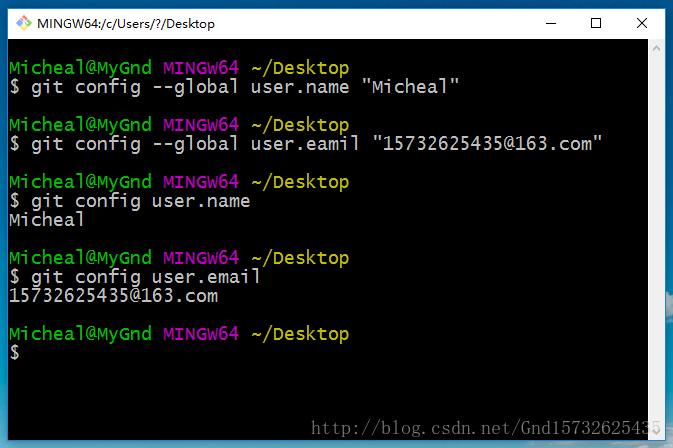
查看当前配置：

$ git config user.name //查看用户名

$ git config user.email //查看用户邮箱地址

$ git config -l //查看所有配置

整体操作的效果图：



# 2.生成自己的公钥和私钥

步骤1

ssh-keygen -t rsa -C "your email address"

步骤2

回车3下

步骤3

cd ~/.ssh

步骤4（这一步是查看公钥，要保留，一会要用）

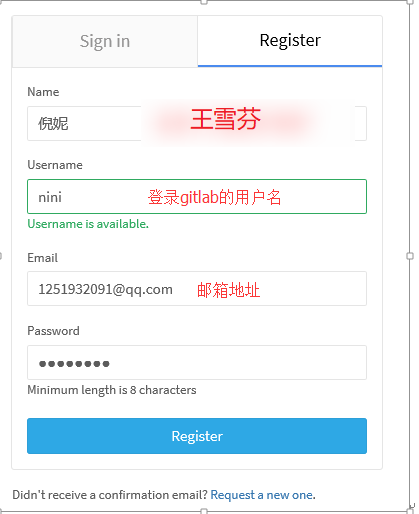
cat id\_rsa.pub

步骤5给开发人员注册gitlab账号

开发人员在自己的电脑上配置好公钥后，根据公钥内容中的邮箱地址，注册gitlab账号（也就是自己在步骤一的邮箱地址），

注册地址

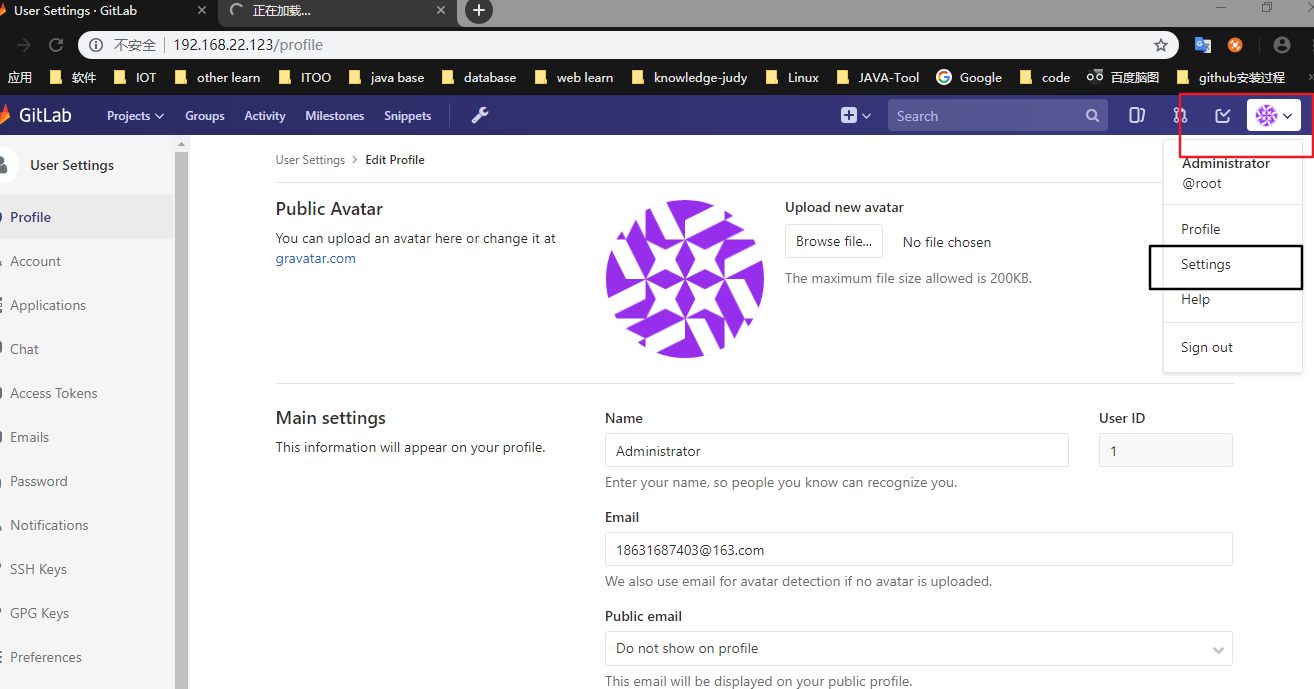
http://git.dmsd.tech/

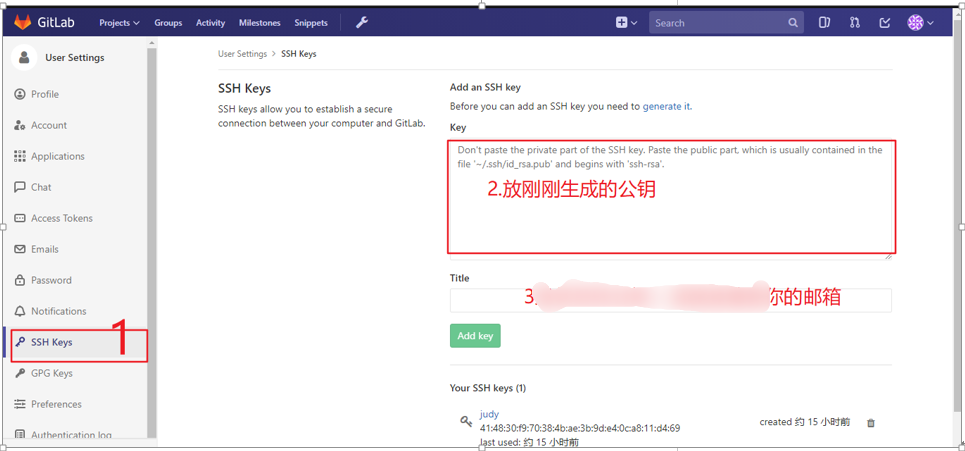


注意：邮箱和用户名需是唯一的。

步骤5防止公钥

使用账号：



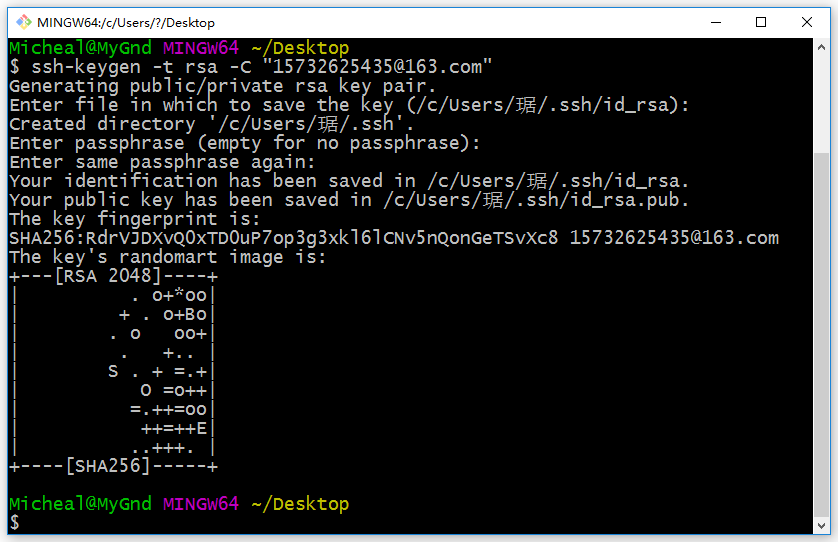


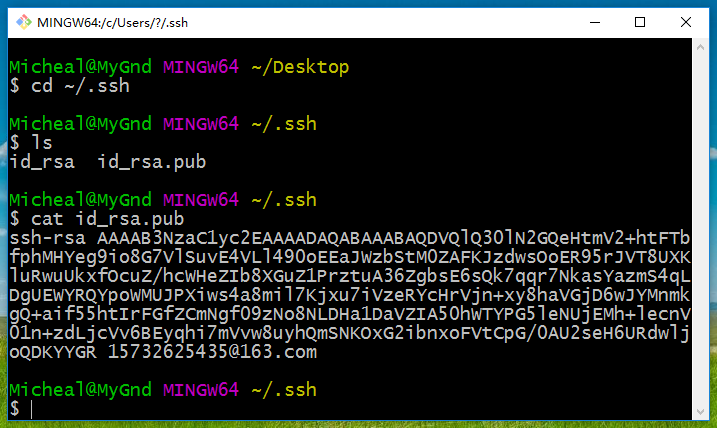
注意：公钥复制的时候复制全

以下是步骤解释

我们从服务器克隆代码的时候，需要一个身份验证。如果将你的公钥放入到服务器，你就而已从上面克隆和推送代码了。

如何生成公钥？在Git Bash中输入命令ssh-keygen -t rsa -C "your email address"，一路回车即可。如下图所示就算成功。公钥放在什么地方？它放用户主目录下的.ssh文件中，里面有id\_rsa和id\_rsa.pub两个文件，这两个就是SSH Key的秘钥对，id\_rsa是私钥，id\_rsa.pub是公钥。你需要注册然后添加到服务上，你才可以克隆代码。



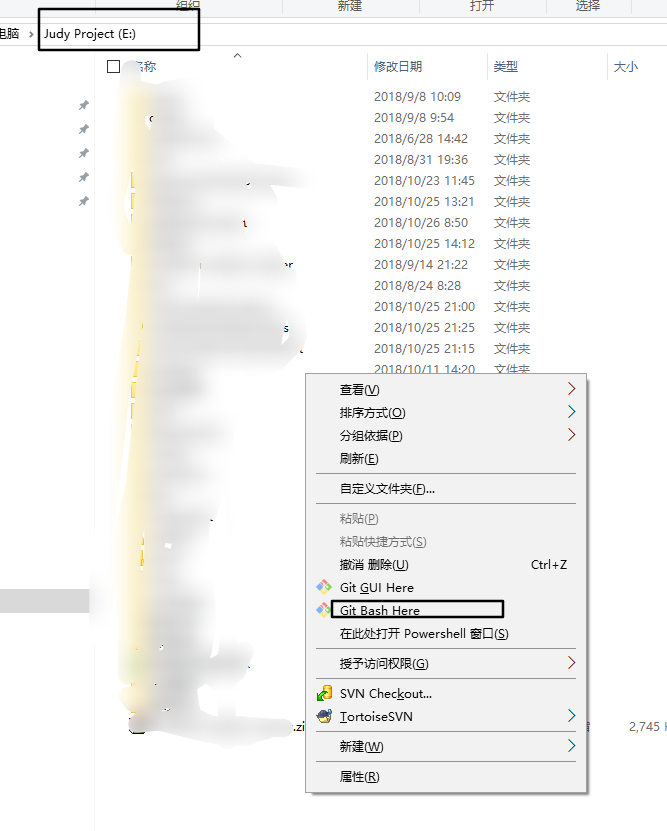


# 3.克隆代码

下面的操作是从gitlab上把代码拉到本地

假如你想在E盘拉取代码，右击点击Git Bash Here

那么代码就到E盘下了，建议新建一个文件夹



从服务器克隆代码到本地的命令是：

$ git clone 代码地址 //把服务器端代码克隆到本地

代码地址

（只是测试）： [git@192.168.22.123:root/ItooTest.git](mailto:git@192.168.22.123:root/ItooTest.git)

1. git add +名字 //从无法到有-----上传本地
2. git commit – m ”注释” //提交到本地缓存区
3. git pull //更新服务器数据避免冲突， -----上传服务器
4. git push //提交到服务器本地文件

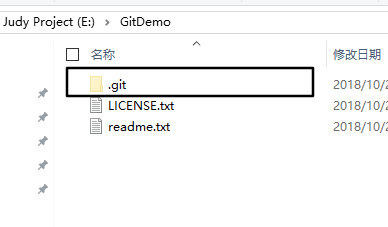
更新代码

Git clone

# 4.创建工作分支

下面的操作是建立工作分支，这个工作分支是dev，操作命令的位置，该位置是你clone下的代码位置，也就是项目里面有一个.git文件，注意：如果没有发现项目里面没有.git 文件。则说明是被隐藏了具体操作：

<https://jingyan.baidu.com/article/e73e26c0f5981124adb6a7ad.html>

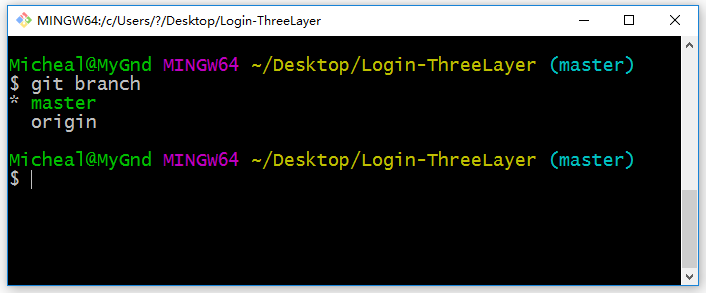


在有.git的目录下，右键，打开Git Bash，

查看本地分支，命令：

第1步操作：

$ git branch



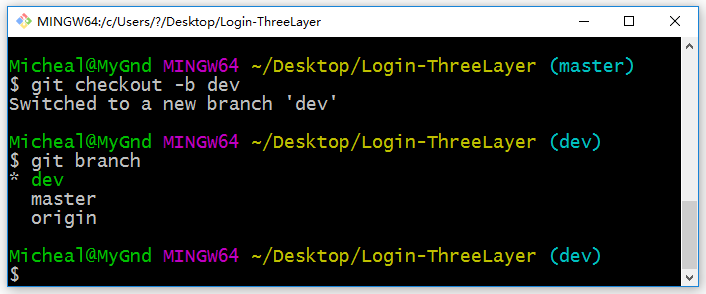
图中的master为本地分支，算是系统分支，自动生成，前面的\*号代表当前所在分支。

第2步操作

我们需要新建我们的工作分支dev，使用命令：

$ git checkout -b dev

$ git branch

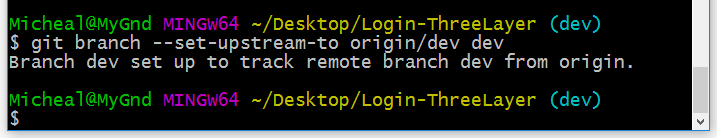


现在我们就建立好了本地的dev分支了。

第3步：

我们需要将我们的本地*dev*分支和远程的dev分支进行关联：

$ git branch --set-upstream-to origin/dev dev



上图表示关联成功！

第4步操作：

4.1创建属于自己的feature分支

从dev分支上检出开发者自己的feature分支，进行开发工作

功能分支的名字：feature-\*(feature-zt)

创建一个功能分支

//从dev分支新建并检出feature分支

$ git checkout –b feature-\* dev



第5步查看当前所在分支

git branch



# 5.开始进行功能开发，提交代码

开始在feature分支下编写代码，并不断地add和commit。

当我们写完负责模块的代码并测试完成后，首先需要将修改添加到本地仓库。步骤为：在有.git文件的目录下右击，点Git Bash Here，输入命令：

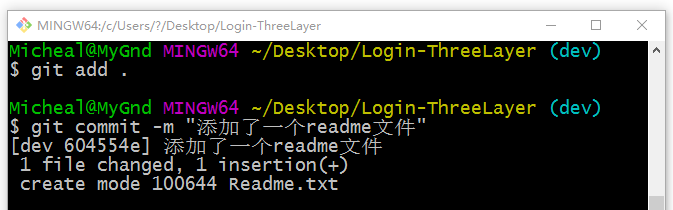
第1步操作

$ git add filename  （git add -a）

//filename为修改的文件名，也可以为点（.）,即全部文件

$ git commit -m “top information” //引号中的为注释

例如：



这样，你的修改就添加到了本地的版本库进行管理了。

第2步操作

这时再切换回dev分支：

$ git checkout dev



更新远程代码，看dev分支是否有更新

$ git pull origin dev



第3步操作

切换为功能分支（可以使用tab键自动补全）

git checkout feature-zt

合并dev分支到自己的功能分支

git merge dev (有冲突，解决冲突，利用git status)

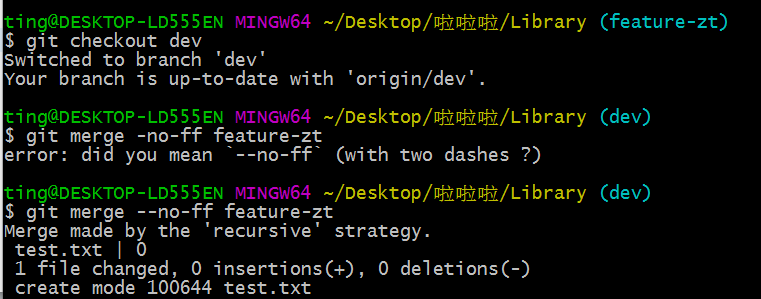
第4步操作

切换为dev分支

git checkout dev

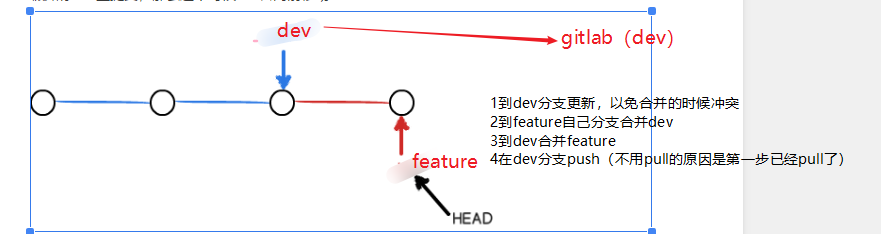
合并feature分支到dev分支

git merge --no-ff feature-zt

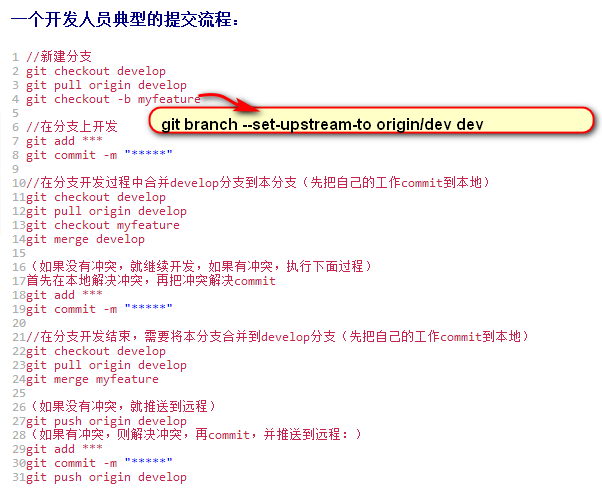
推送dev分支到远端。如下图所示：

git push origin dev





# 6.常用命令

git status //查看文件状态，查看冲突情况

手动解决冲突：

Ising System; 
«<<<<< HEAD 
Ising System. Collections. Generic ; 
Ising System. Collections. Generic; 123 
44fc5149405efddceb4897acb24ca94ea1f4272e 

撤销

git revert HEAD //撤销上一次操作

git revert HEAD^ //撤销上上次操作

git revert commit SHA //撤销指定操作提交

gti reset –hard origin/master //将本地的状态回退到和远程的一样

git reset 057d //回退到某个版本

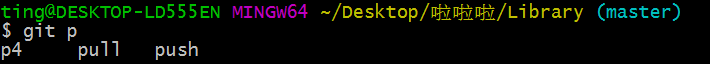
查看提交历史

git log //查看所有提交记录

git log –p -2 //查看最近两次更新记录

//小技巧

1. git co 然后连按两次tab键会看到相关命令提示



2.git命令别名

git config --global alias.co checkout //co为别名，checkout为原名



# 7.Git 简易教程

（建议：手动操作一遍，对于git会更了解）

<https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000>

# 8.Git命令清单

<http://www.ruanyifeng.com/blog/2015/12/git-cheat-sheet.html>