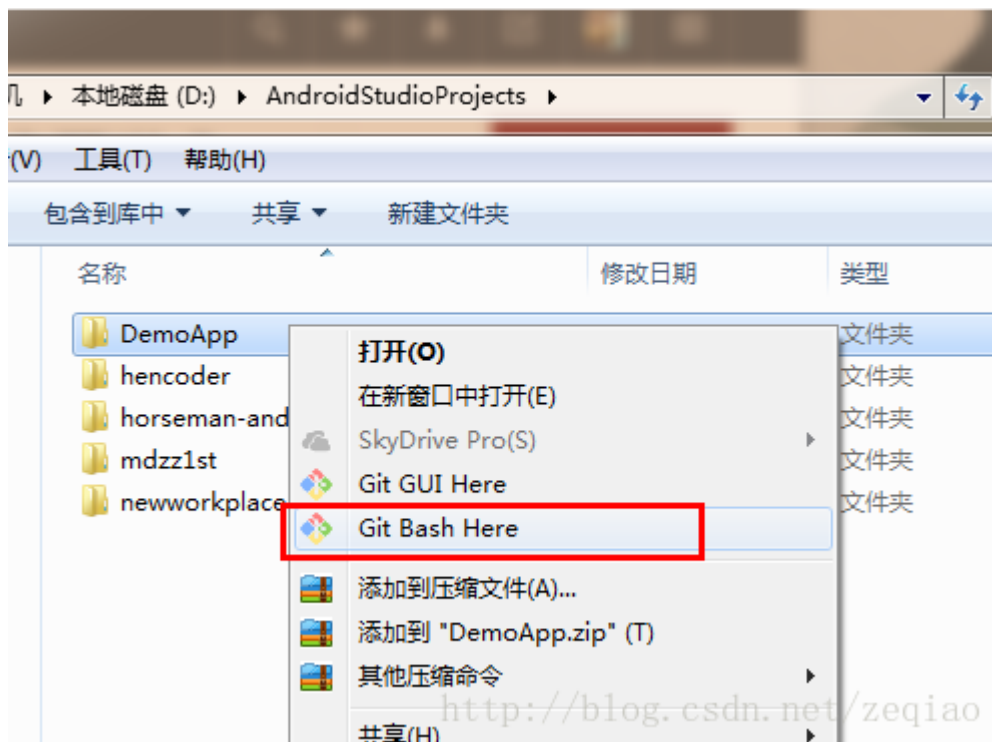


使用 git 进行项目同步开发的步骤

Git 是一款免费、开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目

1 生成 gitlab 的 SSH keys

第一步：打开项目资源路径，右键点击“Git Bash Here”，打开 GitBash



第二步：创建 SSH keys

SSH keys 是什么东西可自己查，原先自己也疑惑，为什么有些教程需要创建，有些又不用呢？检索之后发现，其实并不一定要 SSH keys，百度知道上有一个很好的答案：

“ssh 是一种安全的传输模式，github 要求推送代码的用户是合法的，所以每次推送时候都要输入账号密码，用以验证你是否为合法用户，为了省去每次都要输入密码的步骤，采用 ssh 公钥，密钥；也就是你说的 sshkey 来验证你是否为合法用户，在你的电脑生成了一个唯一的 ssh 公钥和私钥，公钥放到 github 上面，当你推送的时候，git 就会匹配你的私钥是否跟 github 上面的公钥是配对的，正确就认为你是合法的，允许推送。sshkey 可以理解为是你的身份标识，放在 github 上面表明你是这个项目的开发人员，但是别人是可以截获的，你本机的私钥别人就无法截获，sshkey 可以保证每次传输都是安全的。”

在打开的 GitBash 中输入以下命令行，然后一直回车，就会在指定的文件夹中生成密钥：

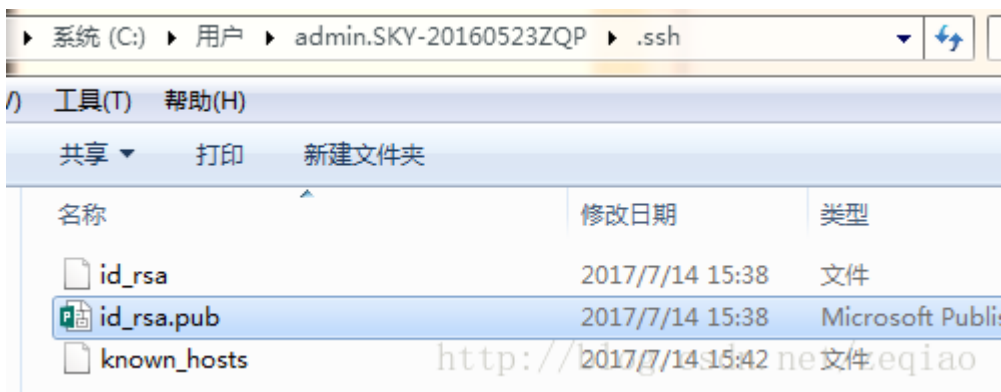
```
ssh-keygen -t rsa -C "你注册 github 时使用的邮箱"
```

然后系统提示输入文件保存位置等信息，连续敲三次回车即可，生成的 SSH key 文件保存在中～
/.ssh/id_rsa.pub

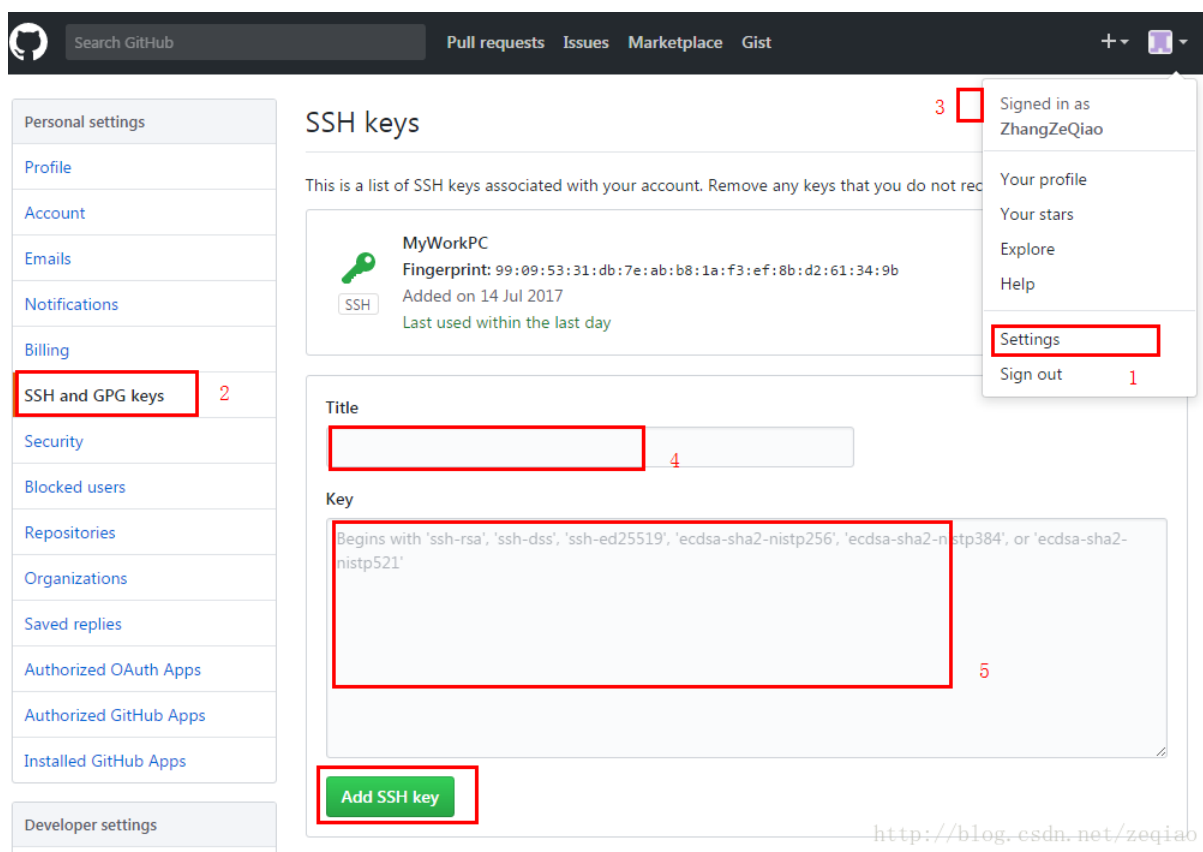
```
MINGW64:/d/AndroidStudioProjects/DemoApp
joe@GZ-PC-0060 MINGW64 /d/AndroidStudioProjects/DemoApp
$ ssh-keygen -t rsa -C "571129524@qq.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c:/Users/admin.SKY-20160523ZQP/.ssh/id_rsa):
Created directory '/c:/Users/admin.SKY-20160523ZQP/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c:/Users/admin.SKY-20160523ZQP/.ssh/id_rsa.
Your public key has been saved in /c:/Users/admin.SKY-20160523ZQP/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:wGhTV/2+C+Jazmi7wWaoCHwVlUXUeE5UNy+NyzG0NZM 571129524@qq.com
The key's randomart image is:
+---[RSA 2048]---+
|      . . . . .==BB|
|      + . . . oo..EX|
|      + . . . oo..EX|
|      + . . . oo..EX|
|      + . . . oo..EX|
|      + . . . oo..EX|
|      + . . . oo..EX|
|      + . . . oo..EX|
|      + . . . oo..EX|
|      + . . . oo..EX|
+---[RSA 2048]---+

http://blog.csdn.net/zeqiao
```

打开对应位置的文件:



将其中的 key 复制到自己的 SSH and GPG keys 中, 新建 key:



验证是否成功:

```
ssh -T git@github.com
```

```
joe@GZ-PC-0060 MINGW64 /d/AndroidStudioProjects/DemoApp
$ ssh -T git@github.com
The authenticity of host 'github.com (192.30.255.113)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.255.113' (RSA) to the list of known hosts
Hi ZhangZeQiao! You've successfully authenticated, but GitHub does not provide shell access.
joe@GZ-PC-0060 MINGW64 /d/AndroidStudioProjects/DemoApp
```

也可以使用命令: `vim ~/.ssh/id_rsa.pub` 或 `more ~/.ssh/id_rsa.pub`

2 克隆远程仓库项目

使用命令 : `git clone git@gitlab.xxxxxxxx.cn:customer/project.git`

3 创建本地分支 dev

使用命令: `git checkout -b dev`

4 添加需要提交的文件

使用命令：

```
git add *
```

添加需要提交的文件

5 添加提交描述

使用命令：

```
git commit -m '提交描述'
```

用于提交文件

6 切换到 master 分支

使用命令：

```
git checkout master
```

切换分支到 master 主干上，等会合并 dev 分支和远程 origin/master 分支内容

7 pull 主分支内容

使用命令：git pull 合并 origin/master 分支内容

8 合并 dev 分支内容

使用命令：git merge dev 合并 dev 分支内容

9 遇到冲突修改后，再次使用 4 和 5 步骤提交冲突文件

10 提交本地仓库内容至远程仓库

使用命令：

```
git push -u origin master
```

提交本地仓库内容至远程仓库

11 备注：

设置提交用户邮箱，必须与 gitlab 邮箱一致，否则提交不成功

```
git config --global user.name "Your Name"
```

```
git config --global user.email you@example.com
```

参考资料:

<http://www.kancloud.cn/thinkphp/git-cheat-sheet/39645>