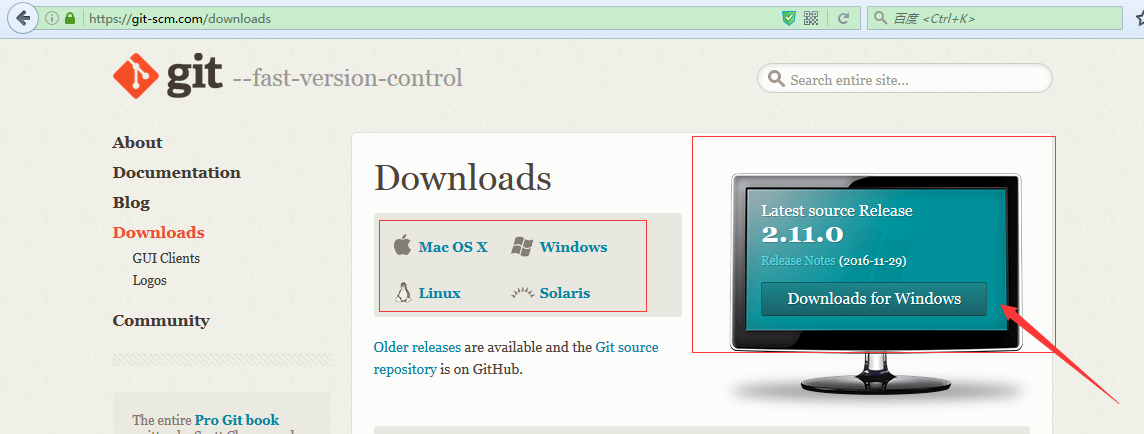
git快速入门

注：以本次项目为例，快速参与项目病协作开发。

# 1 安装Git（windows）

在git官网 <https://git-scm.com/downloads> 下载最新的git安装包(注意有32位安装白和 64位安装包),下载完成以后，带点击exe文件进行安装，一直点下一步直到完成安装。





# 2 配置git

因为Git是分布式版本控制系统，所以，每个机器都必须自报家门：你的名字和Email地址。

配置git，就相当于给git一个身份证号，用来区分git仓库。

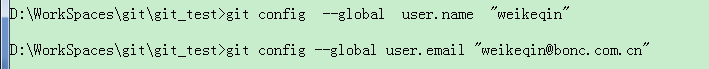
安装完成后，还需要最后一步设置，在命令行输入：

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

$ git config --global user.email "email@example.com"

例如：git config --global user.name "Your Name"

git config --global user.email "email@example.com"



注意git config命令的--global参数，用了这个参数，表示你这台机器上所有的Git仓库都会使用这个配置，当然也可以对某个仓库指定不同的用户名和Email地址。

# 3 使用git

## 3.1 生成并配置SSH KEY

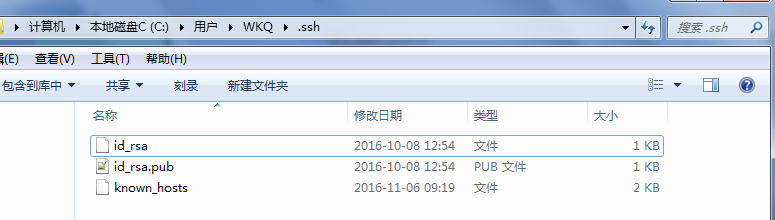
在安装git的目录下，进入git-bash.exe，然后输入：

ssh-keygen -t rsa -C ["weikeqin@bonc.com.cn"](mailto:%22weikeqin@bonc.com.cn%22) 用的时候记得邮箱写自己的

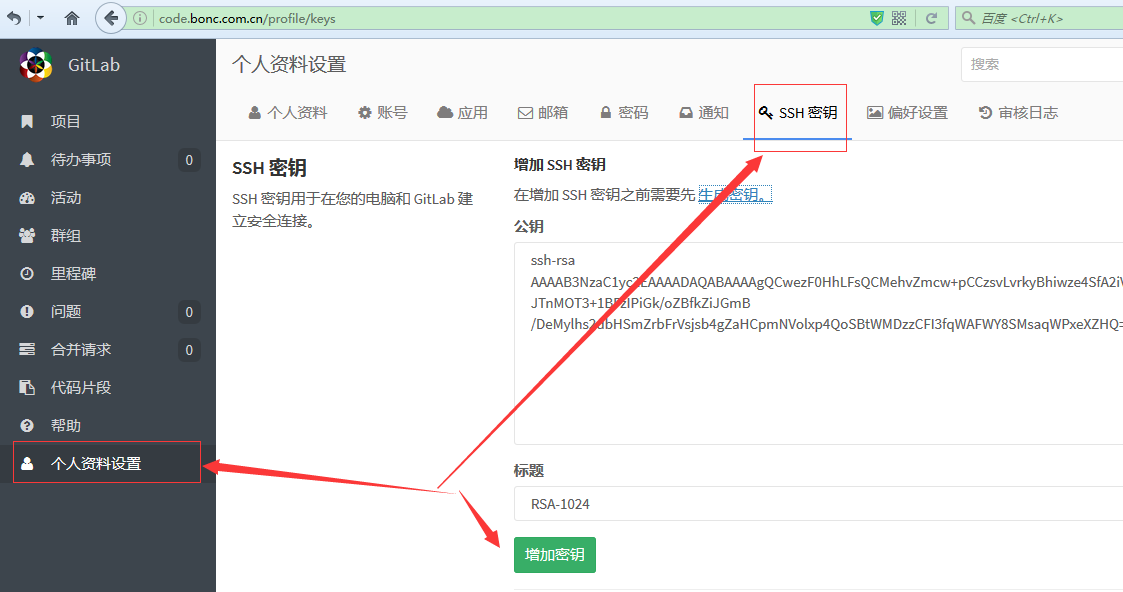
按3个回车，密码为空

生成SSH KEY 以后，会保存在系统盘当前用户的.ssh目录下。

Id\_rsa 是密钥 is\_rsa.pub是公钥，我们需要把公钥添加到git服务器上



Gitlab 添加SSH KEY 把is\_rsa.pub里的内容复制到公钥框里就可以了



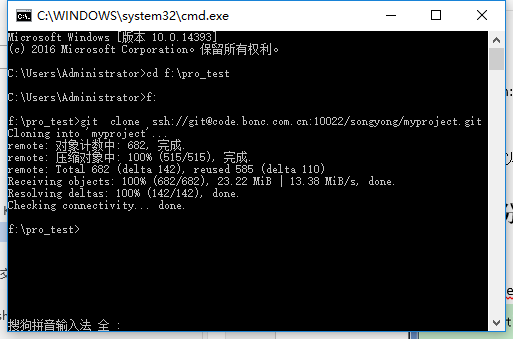
## 3.3 克隆项目

创建一个该项目的工作环境（文件夹）：比如F:\ pro\_test

cmd中cd到工作目录，如：cd F:\pro\_test

本项目SSH：ssh://git@code.bonc.com.cn:10022/songyong/myproject.git

git clone ssh://git@code.bonc.com.cn:10022/songyong/myproject.git



克隆完，就可以在我本地的 F:\ pro\_test下找见myproject目录



## 3.2 创建分支

进入项目文件，cd myproject

git branch dev\_test 创建新分支dev\_test



git branch 查看分支



git checkout dev\_test 切换到分支dev\_test



git status 查看仓库状态



本次项目发布版是在分支master，个人分支可以使用dev\_sy（自定义）

请切换到自己的分支下进行修改提交

## 3.4 对文件修改后提交

可以设置工作区（Working Directory）和版本区（Repository）

工作区（Working Directory），就是经常修改文件代码的文件夹，版本区可以是克隆的项目文件夹，建议刚开始使用工作区和版本区独立使用，减少不必要的反复工作。

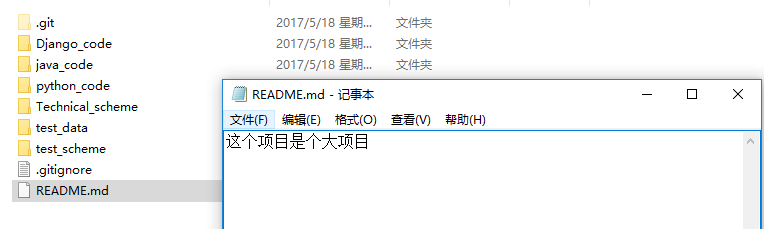
（详见Technical\_scheme\git教材）

示例，1、在当前项目想修改README.md文件，修改其中的内容；2、增加一个文件夹名字为test01，test01下面有一个test01-1的txt文档。

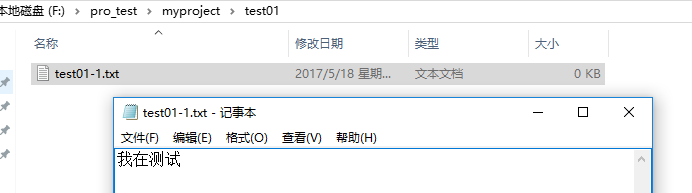
首先增加相关的文件夹或文档。



修改README



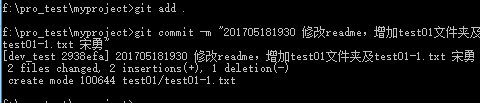
添加test01文件夹，并在文件夹中添加test01-1.txt，并修改内容。



下面是如何提交如何推送。

第一步：是用git add把文件添加进去，实际上就是把文件修改添加到暂存区；

第二步：是用git commit提交更改，实际上就是把暂存区的所有内容提交到当前分支。



第三步：使用 git push 命令把修改提交到远程仓库

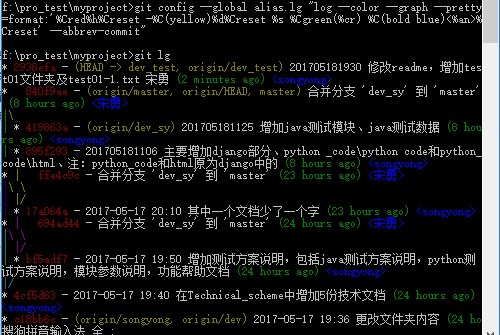


可以看到，把本地的修改提交到远程仓库了。

使用git lg 命令查看修改历史

lg是配置的别名，具体配置如下

git config --global alias.lg "log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit"



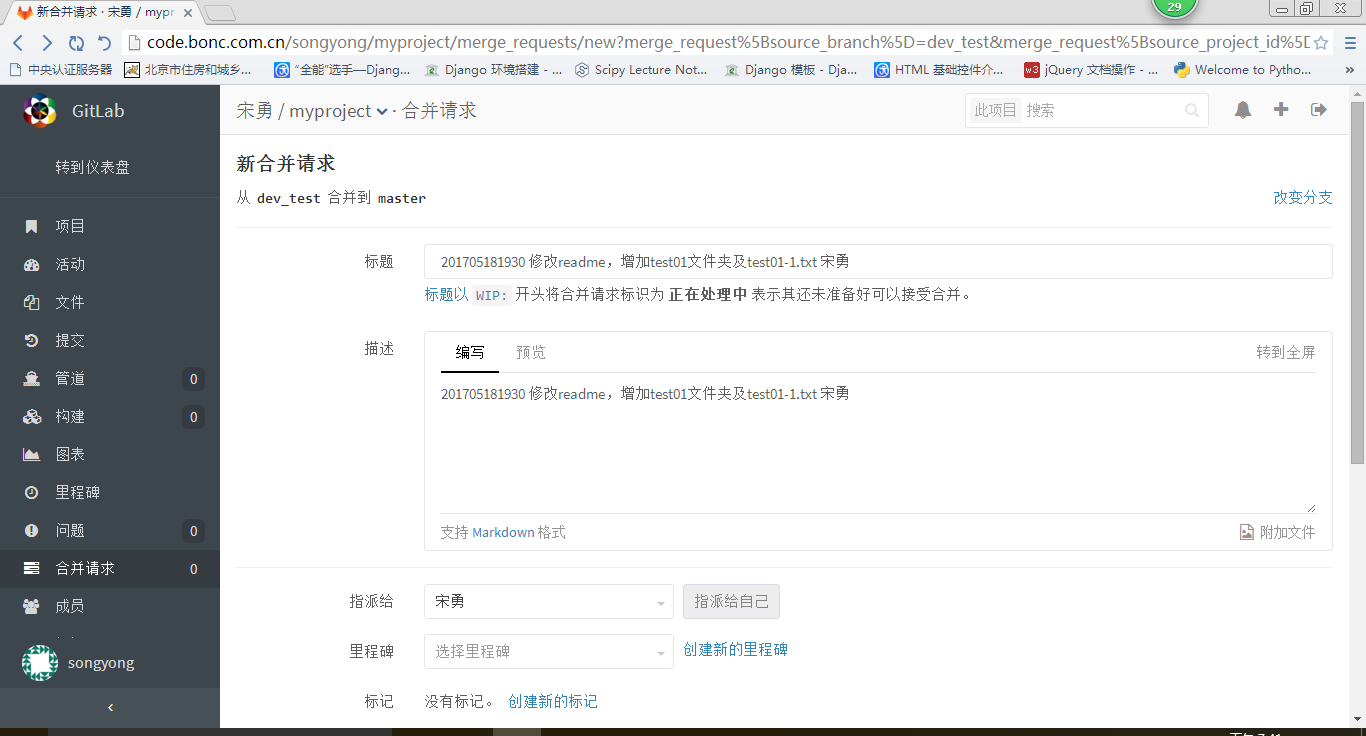
现在只是从本地的dev\_test分支推送到服务器dev\_test分支，需要在服务器把dev\_test分支的内容合并到master分支

第四步：与master合并

在gitlab上可以看到当前的活动：

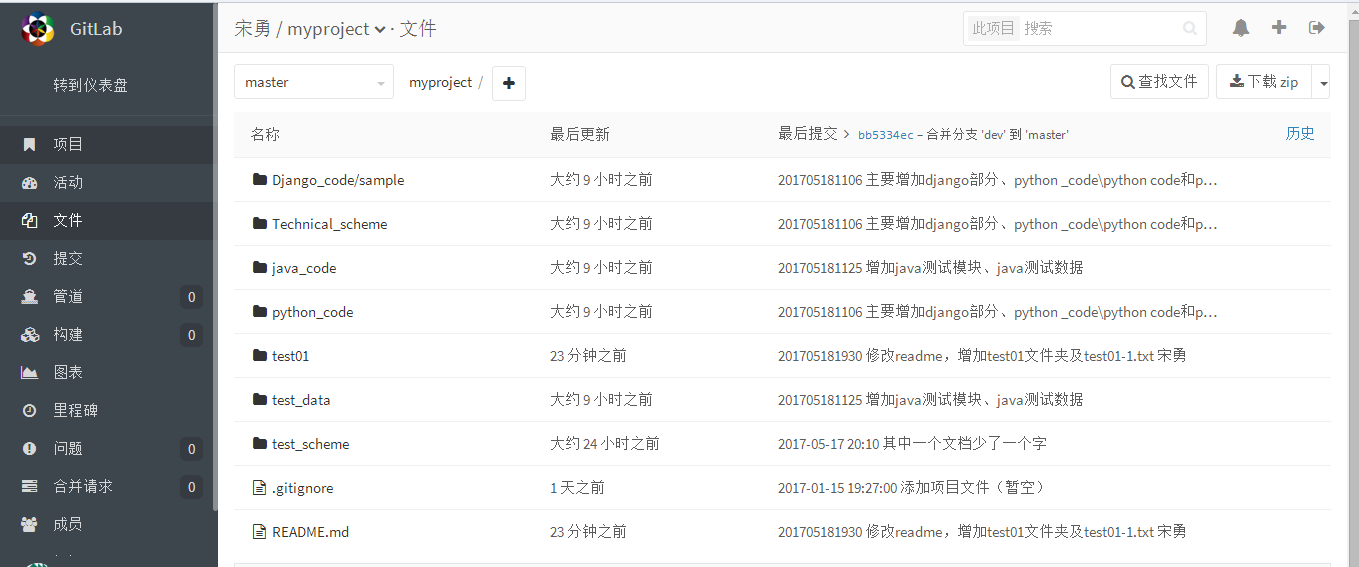


点击创建合并请求，编写标题、描述并指派给自己



点击提交新的合并请求，最后点击接受合并请求。

在文件中，master分支即能看到刚更新的文件和文件夹。



每次如果需要使用到其他人写的代码或者文档，可以先更新项目，然后再提交。

更新项目前需要查看本地仓库状态，如果本地仓 库中有未提交的，或者未push的，可以push或commit后再更新。



更新项目：

git pull origin master

## 3.5 撤销及回退

git pull origin master

## 3.6 pycharm使用ssh