# 使用情况

# GitLab辅助工具安装和介绍：

Git：

下载网址<https://gitforwindows.org/>

TortoiseGit【选择系统对应的位数32或者64】：

下载网址<https://tortoisegit.org/download/>

如上2个软件默认安装即可，安装完成后，右键桌面如下所示：



上图包含Git关键字的选项，可以进行界面操作GitLab，用于上传，下载和提交等操作；

如上Git工具是辅助作用，这些操作基于GitLab系统中已经存在项目，如下讲述如何在GitLab中创建project和权限问题；

# GitLab公网地址：

<http://218.240.53.52:10080/>

# GitLab系统使用

GitLab中创建一个工程，在此工程下存放代码，管理代码；

创建工程分2种：

一是：创建自己的工程供自己使用，同时也可以分享给其它人或者组；

二是：基于组来创建工程，就是先创建一个组，然后在此组里创建相关联的工程；

## 权限说明

（1）Guest 可以提供给测试部，用于提交问题；

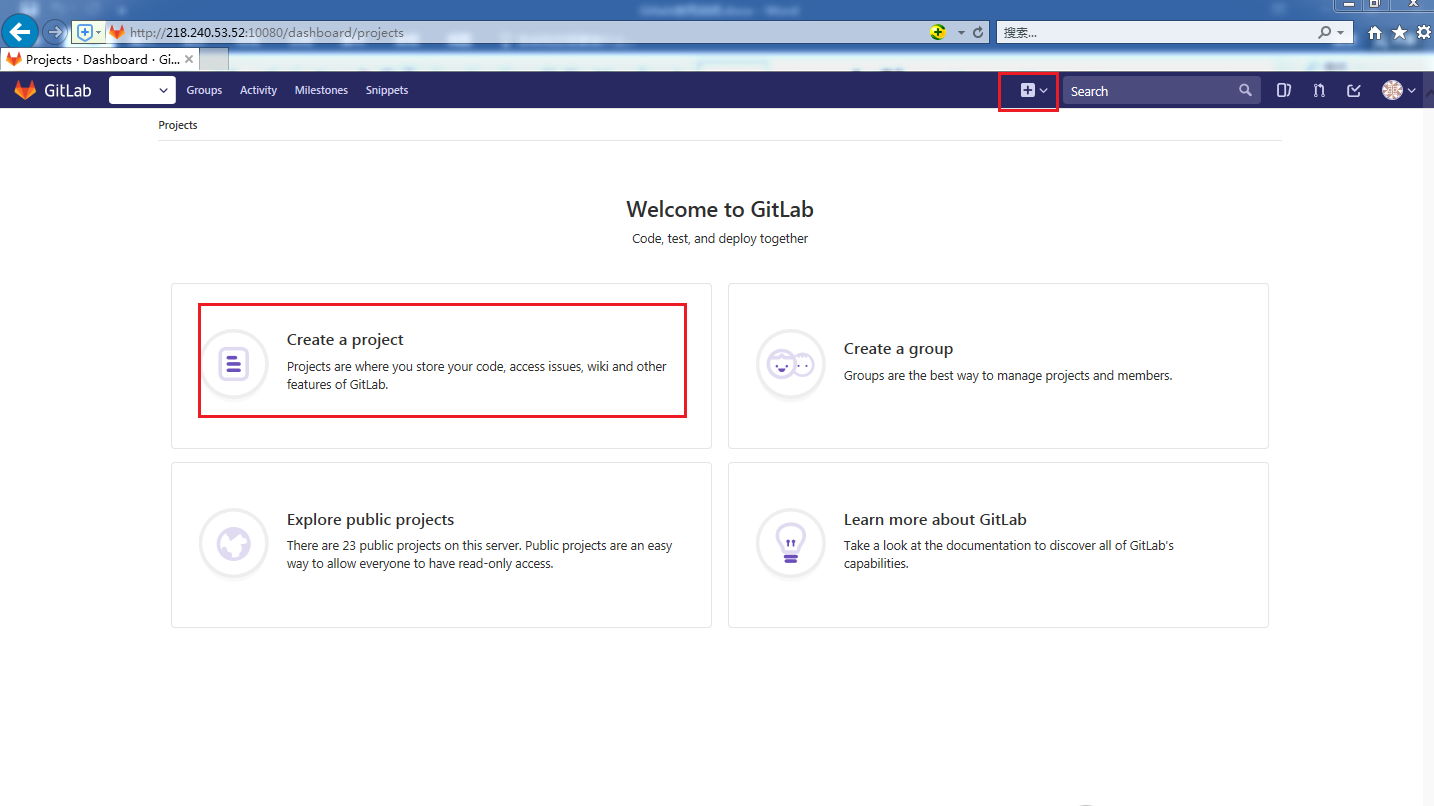
（2）Reporter 可以clone（下载）工程，但不能提交；

（3）Developer可以clone工程，但是push提交时，只能提交到非保护的分支，然后从GitLab网站系统上请求合并到主分支；

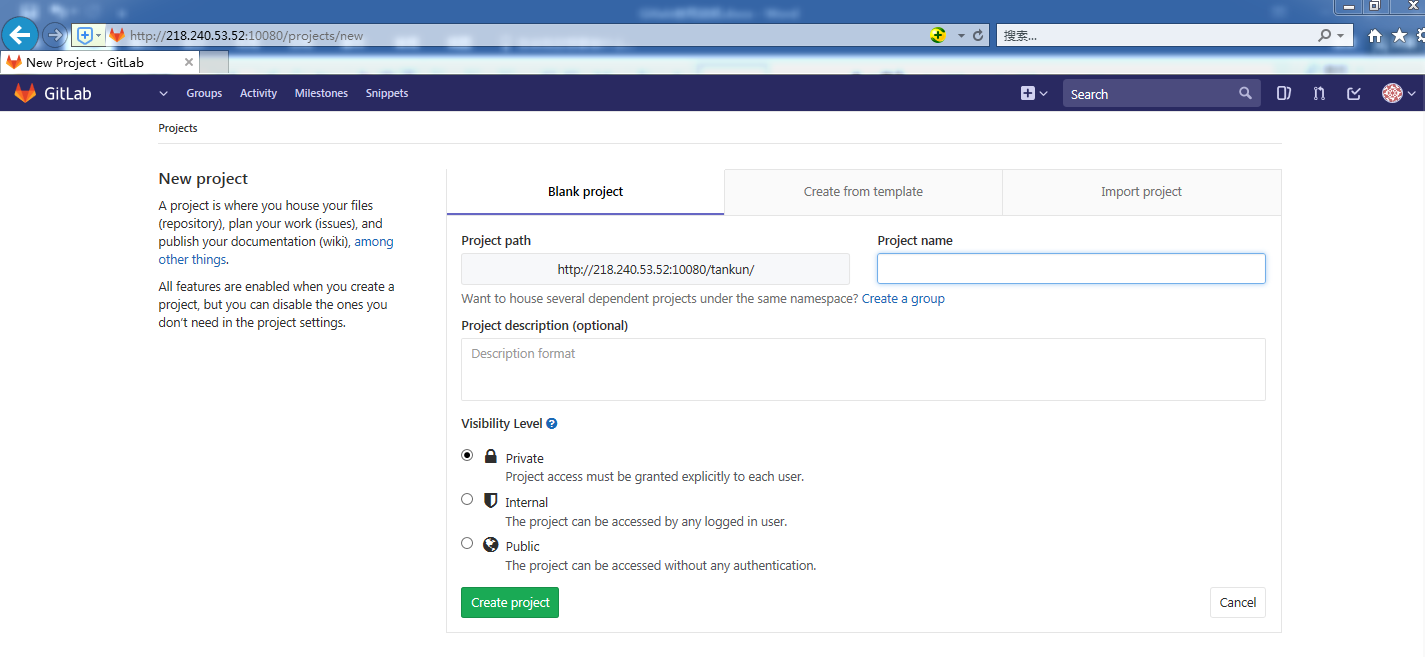
（4）Master 可以clone工程，可以提交所有分支。负责处理Developer的合并请求，并在Review之后合并到主分支或者修改或者拒绝；

## 第一种：创建自己的工程以及权限分配

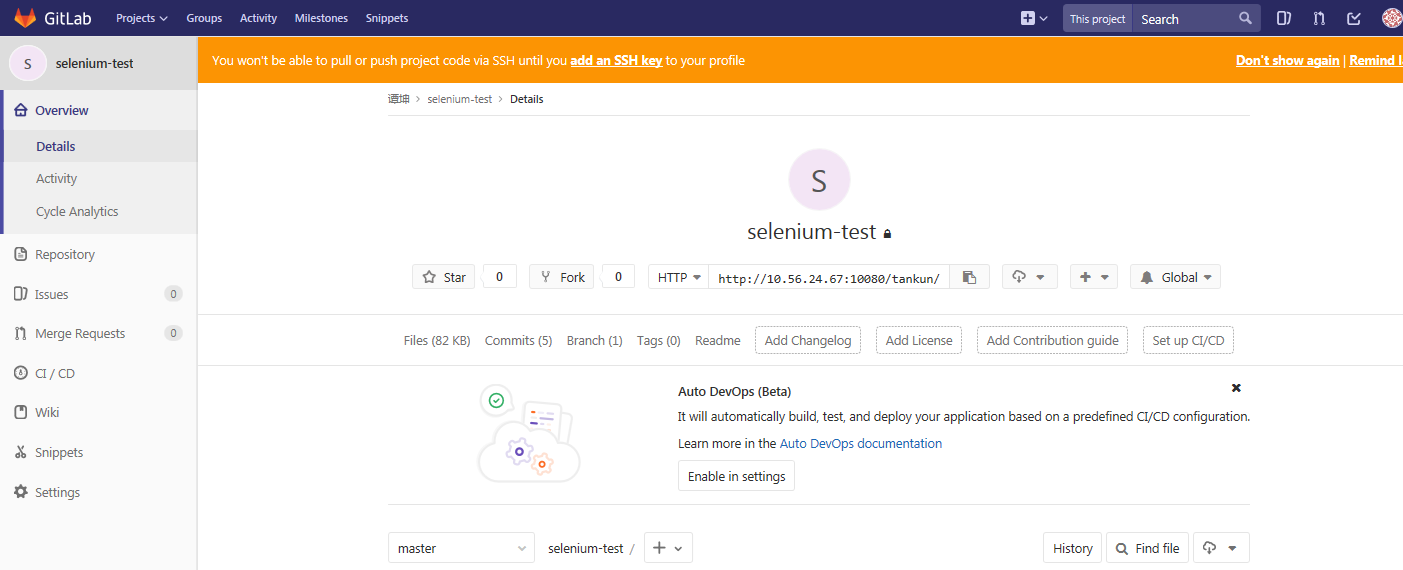
GitLab界面第一次登陆会提示更改密码，更改完密码后会进入如下首页



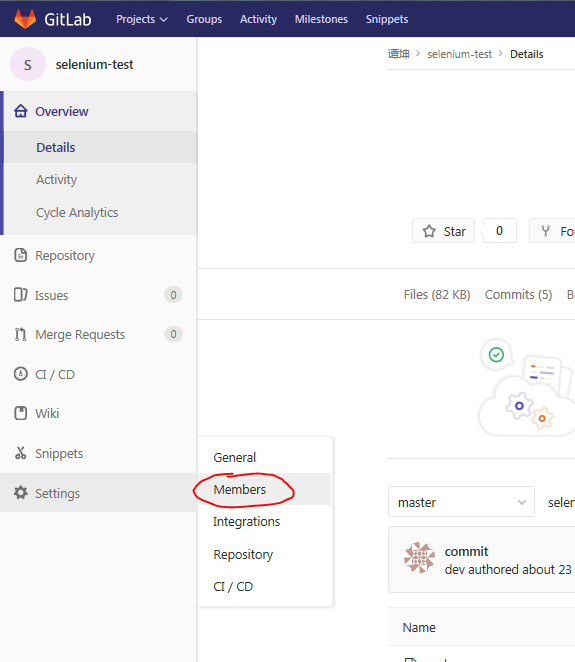
第一次登陆的话，直接点击首页的Create a project；如果不是第一次，点击页面最上面菜单栏中的＋，再点击New project，跳转如下页面：



如上Project name中输入工程名字如selenium-test，Visibility Level选择Private私有的，如果所有人都可见则选择Public，点击Create project

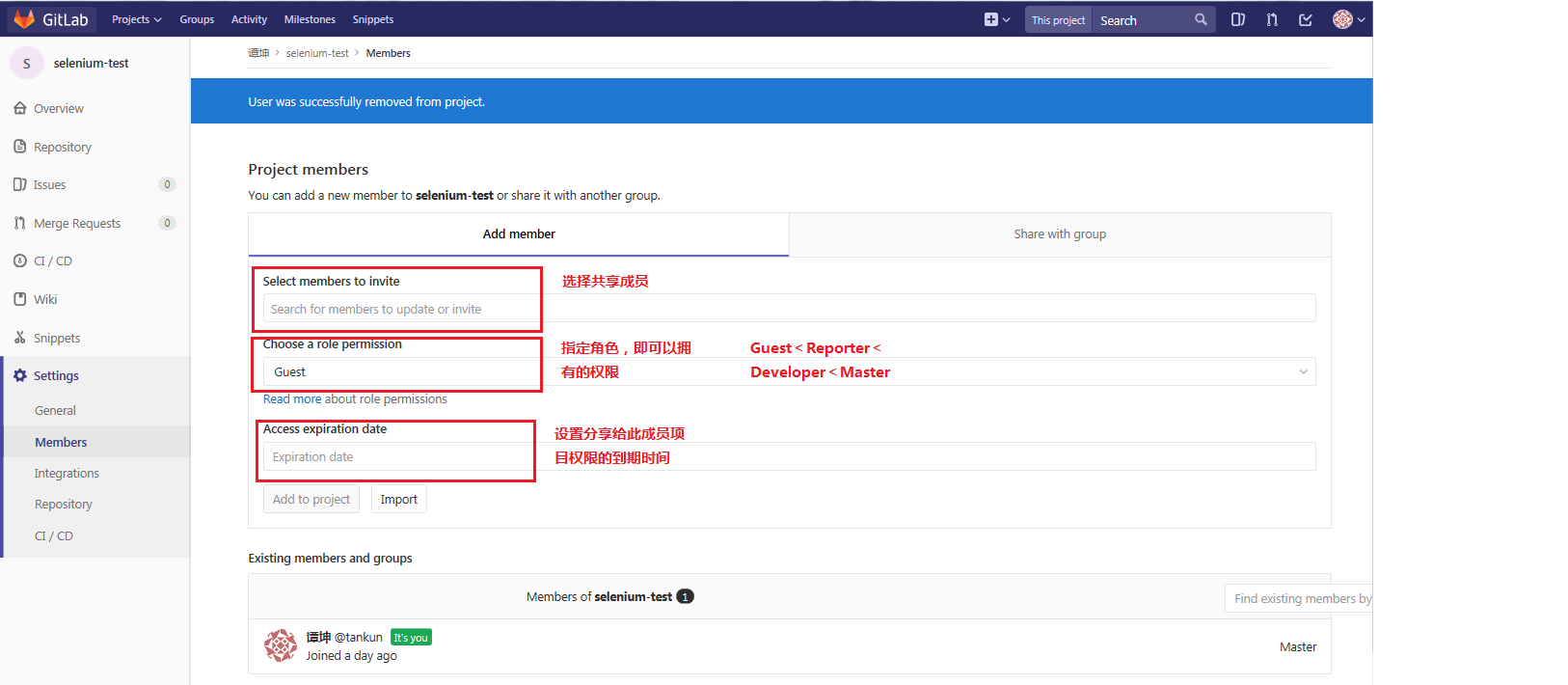


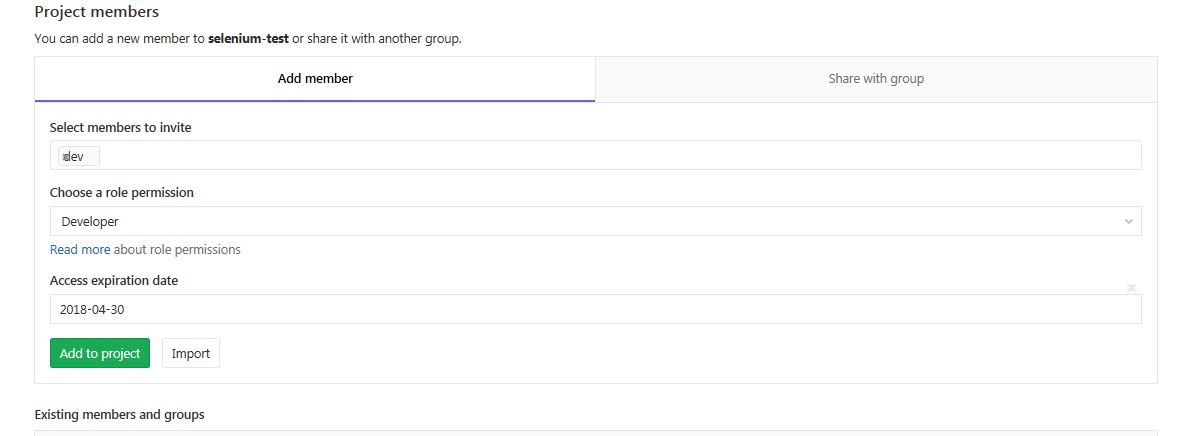
自己创建的工程，自己的角色就是master，拥有最大权限，此工程可以自己独有，也可以分享给其它人；

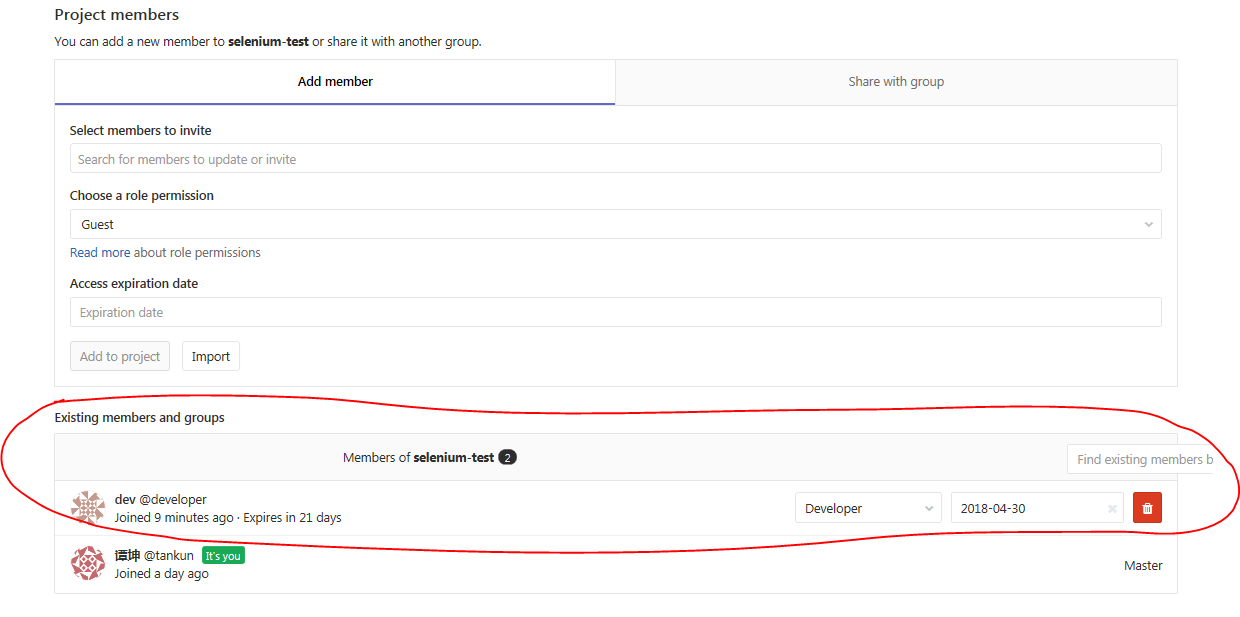


点击Members，可以分享此工程，即此工程可以共享的成员或者组，同时也设置了权限；

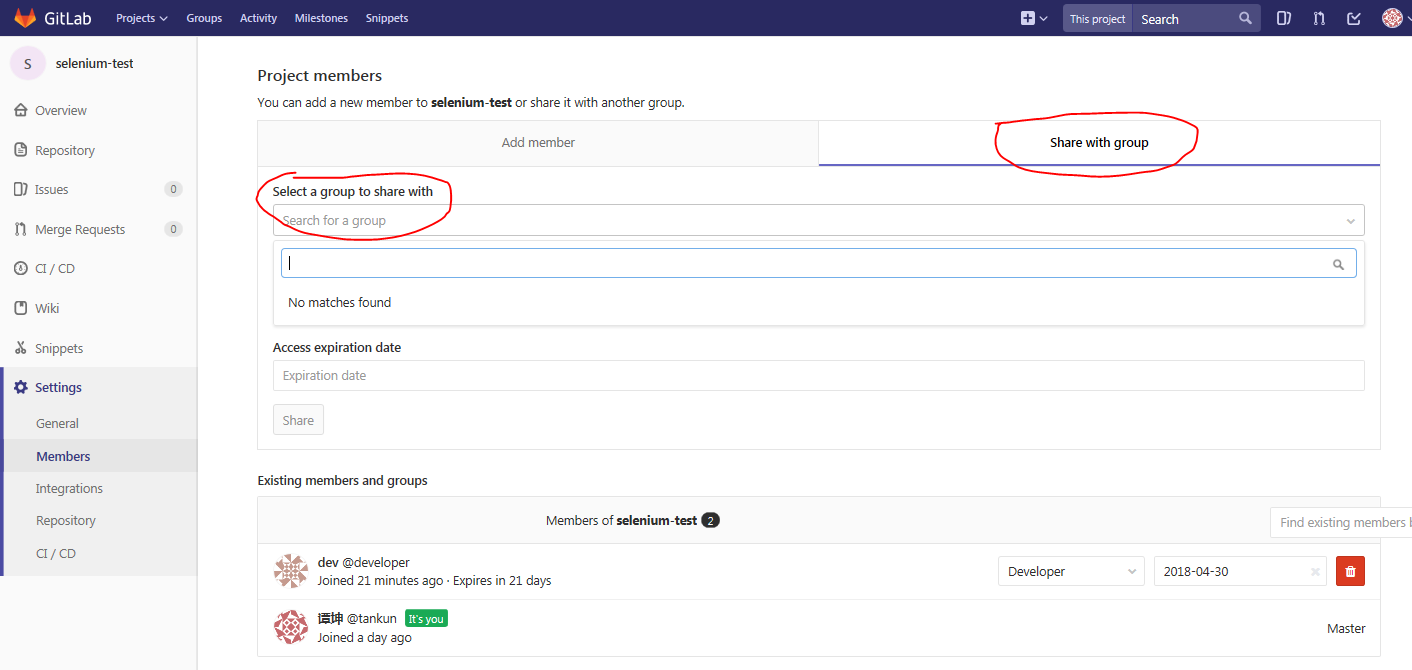
分享给指定的成员：





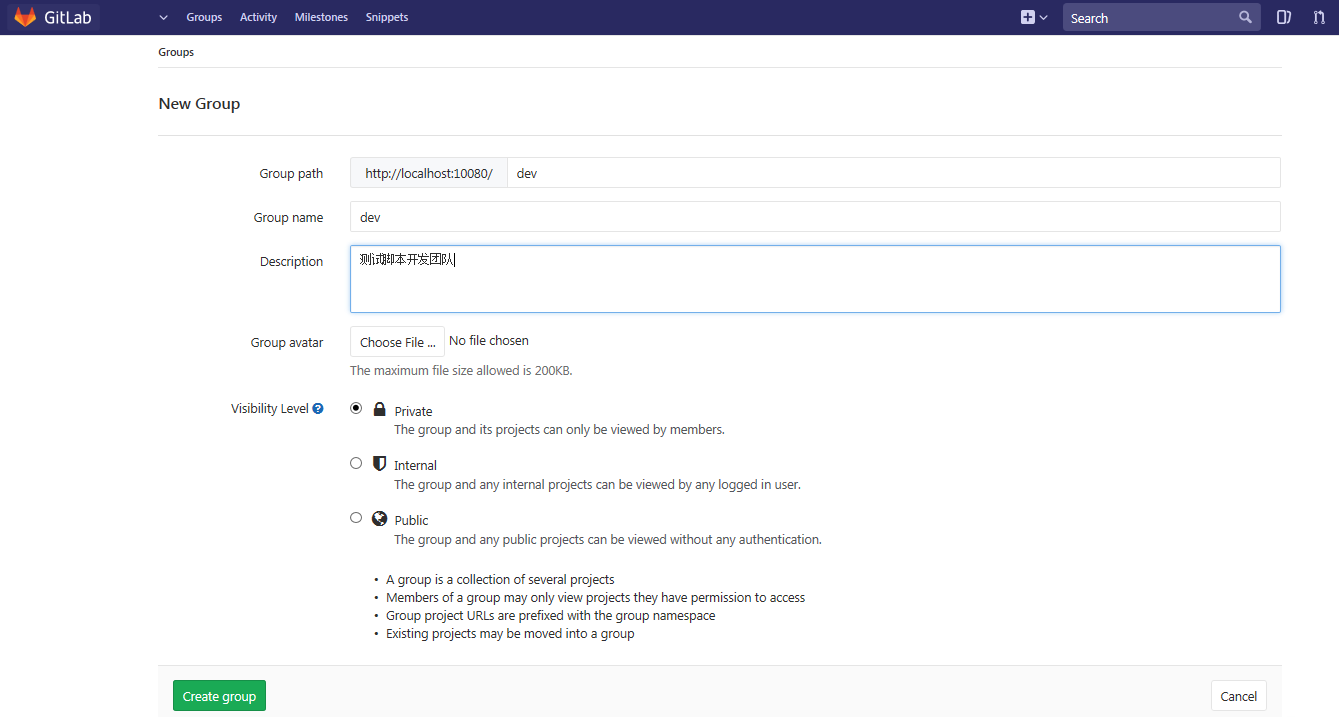


分享给指定的组：这里不讲述，同分享给指定的人员一样的操作，但是前提是有Group存在；

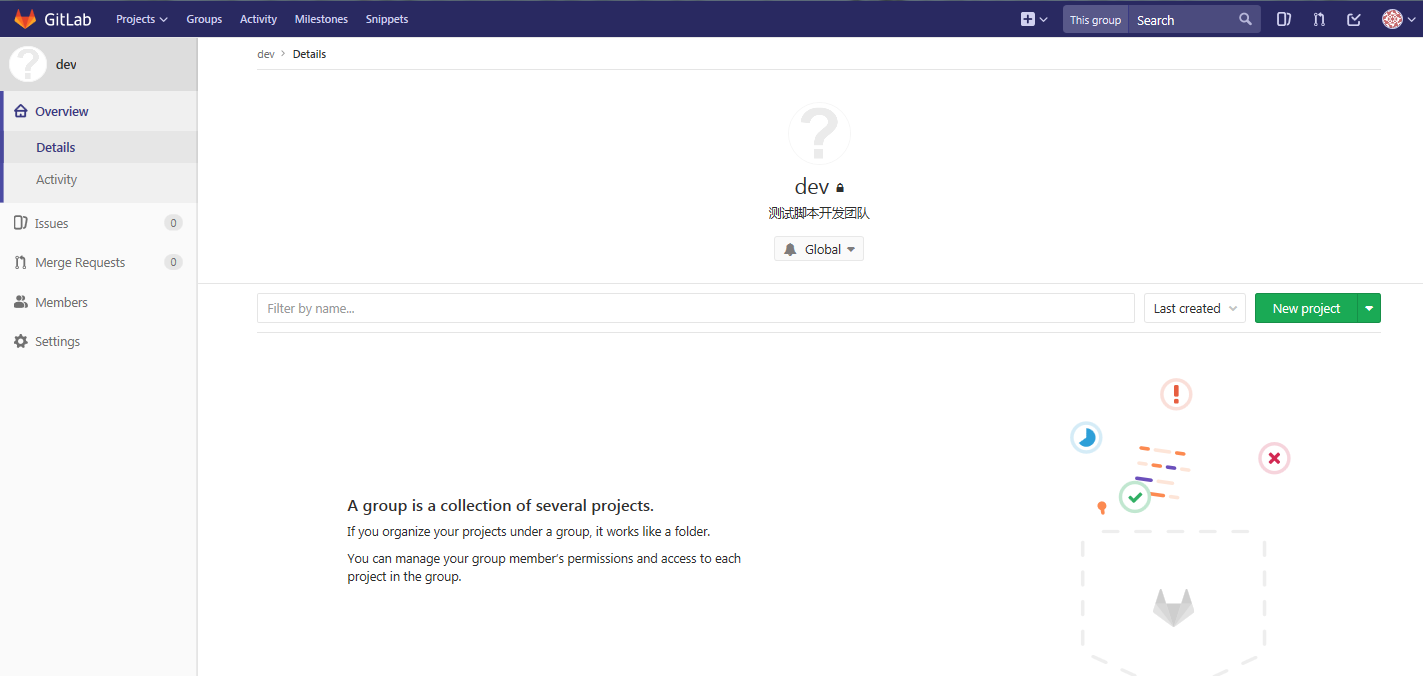


## 第二种：创建基于组的工程以及权限分配

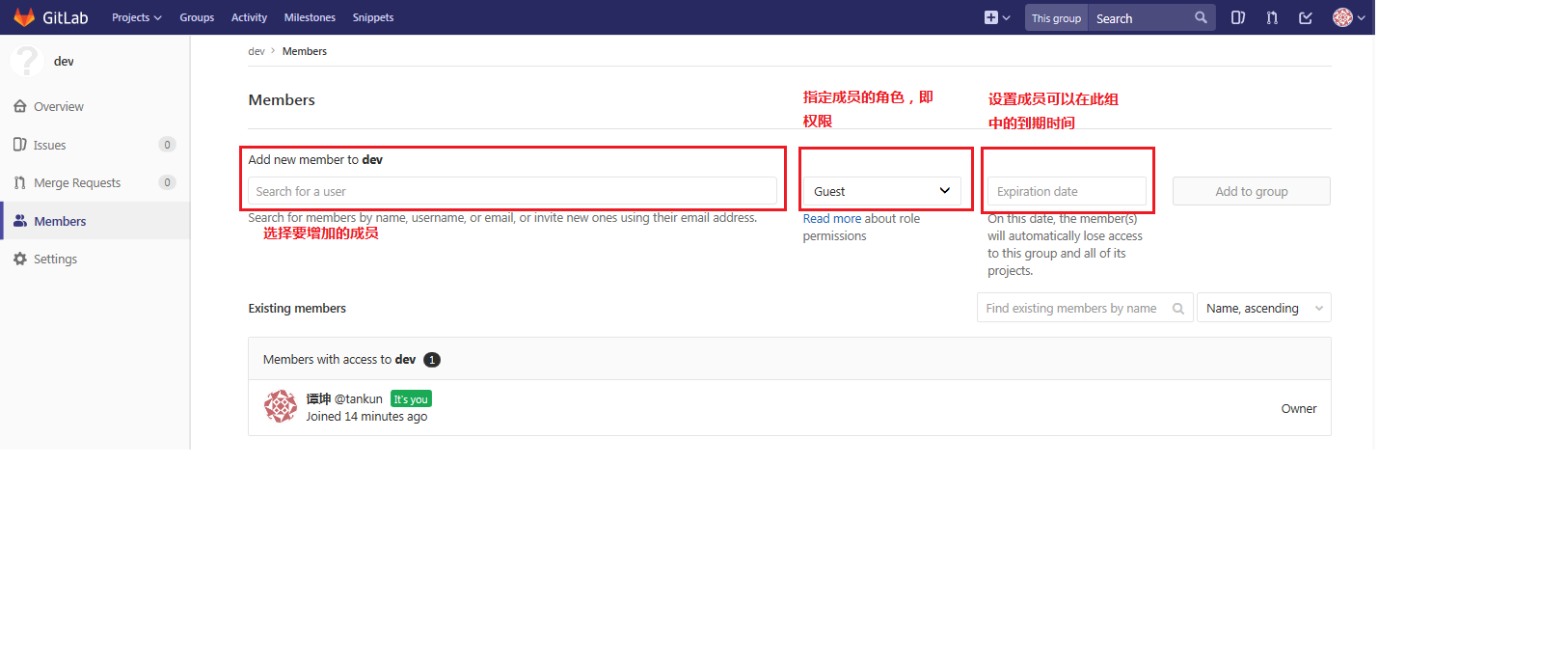
点击页面最上面菜单栏中的＋，再点击New Group，跳转如下页面：

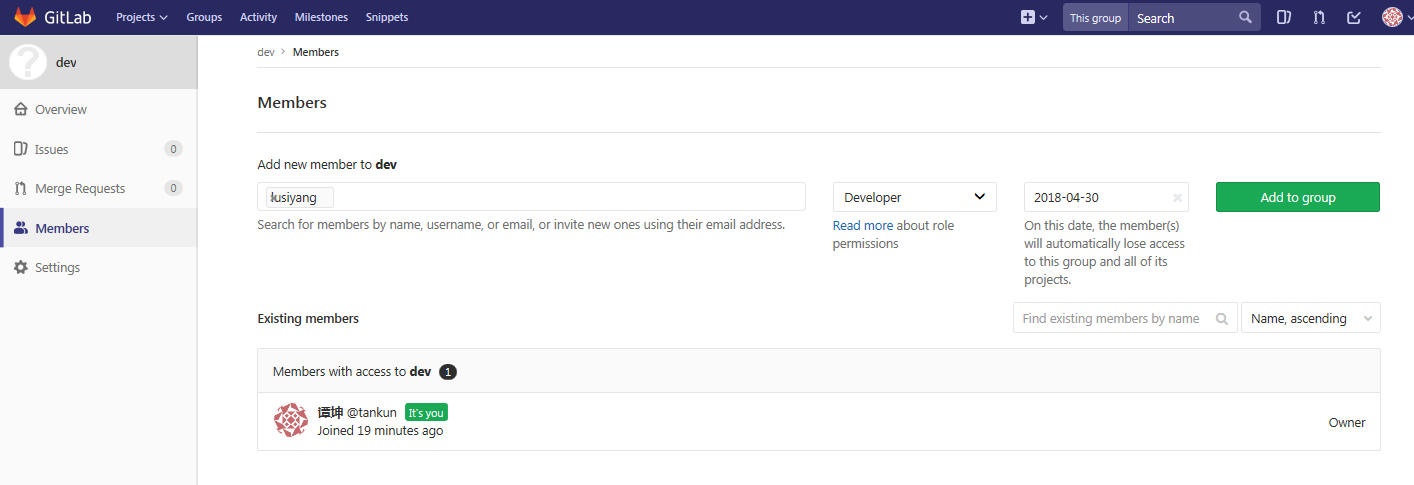


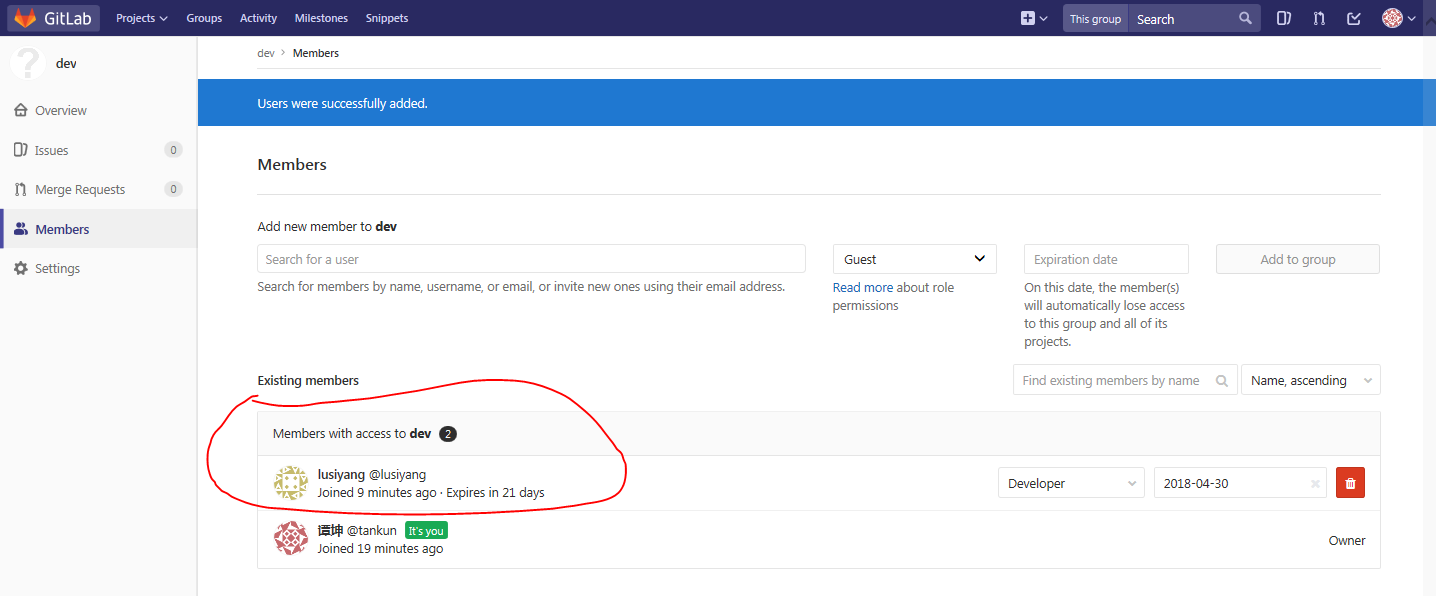
Group path中输入组的目录名，同时输入Group name，还要描述Description，点击Create group：



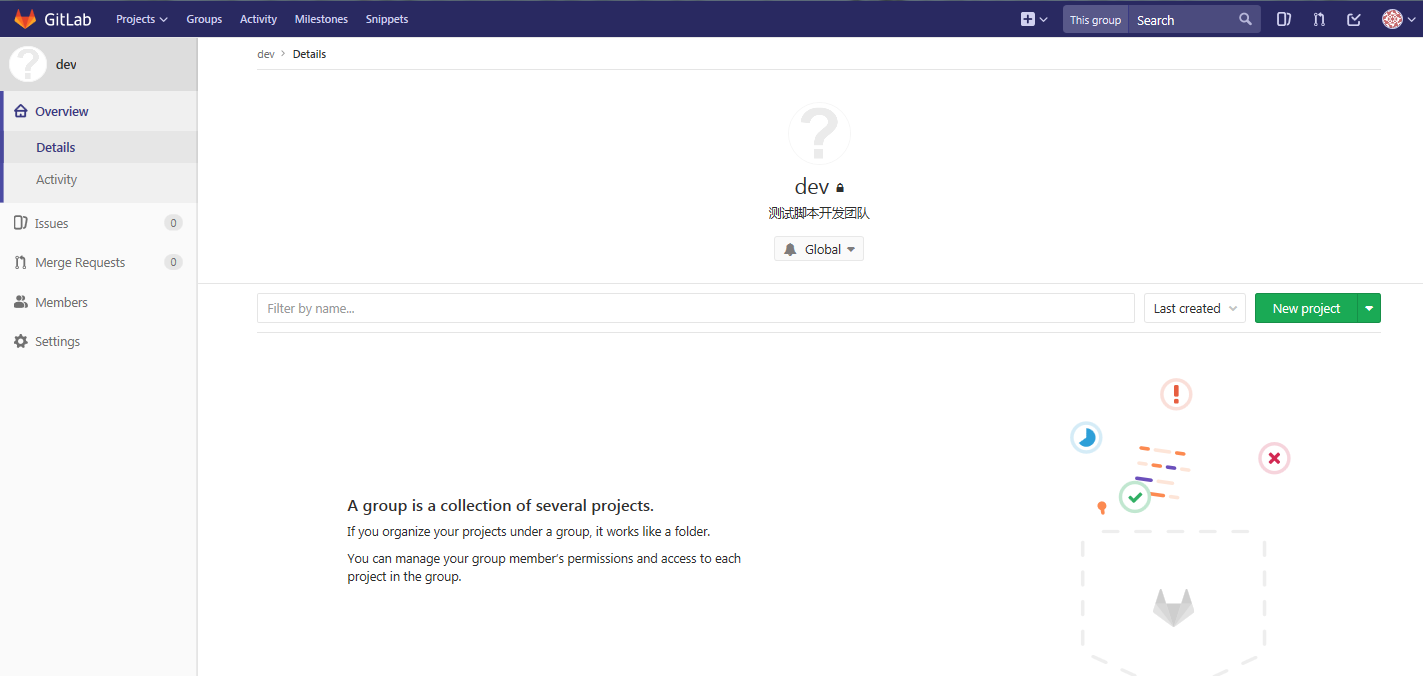
给组增加成员，点击Members,设置相关信息，点击Add to Group：

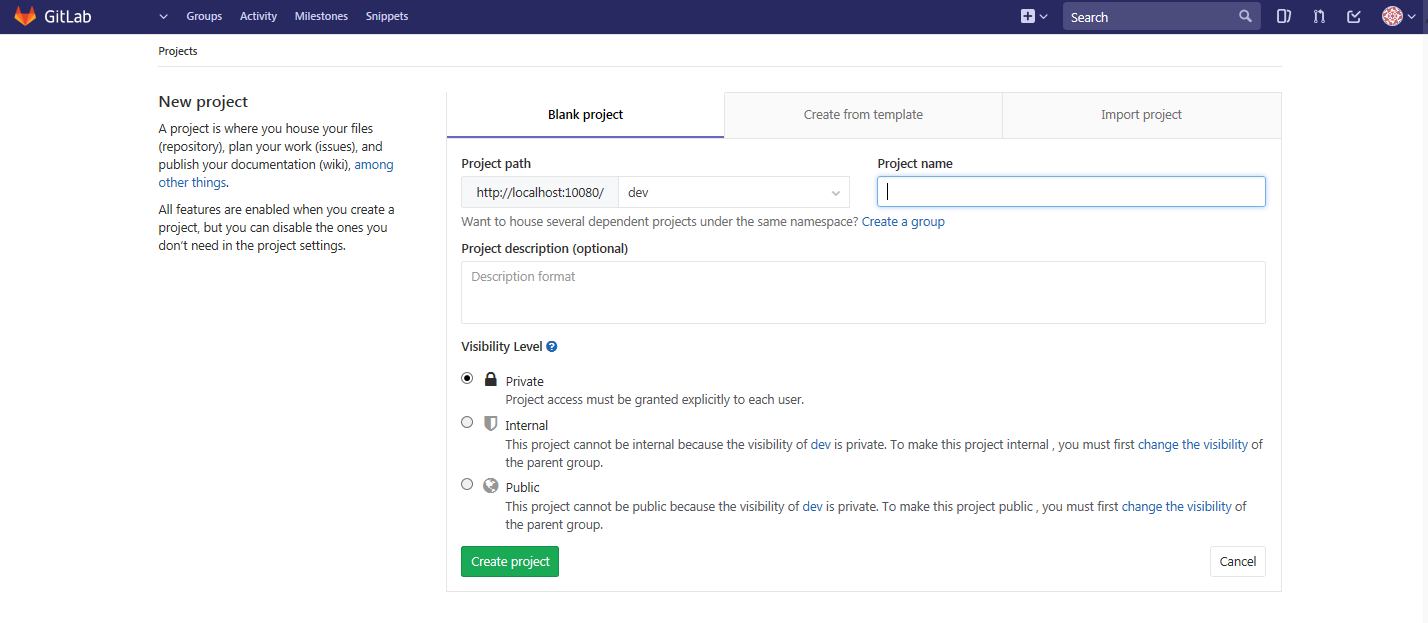




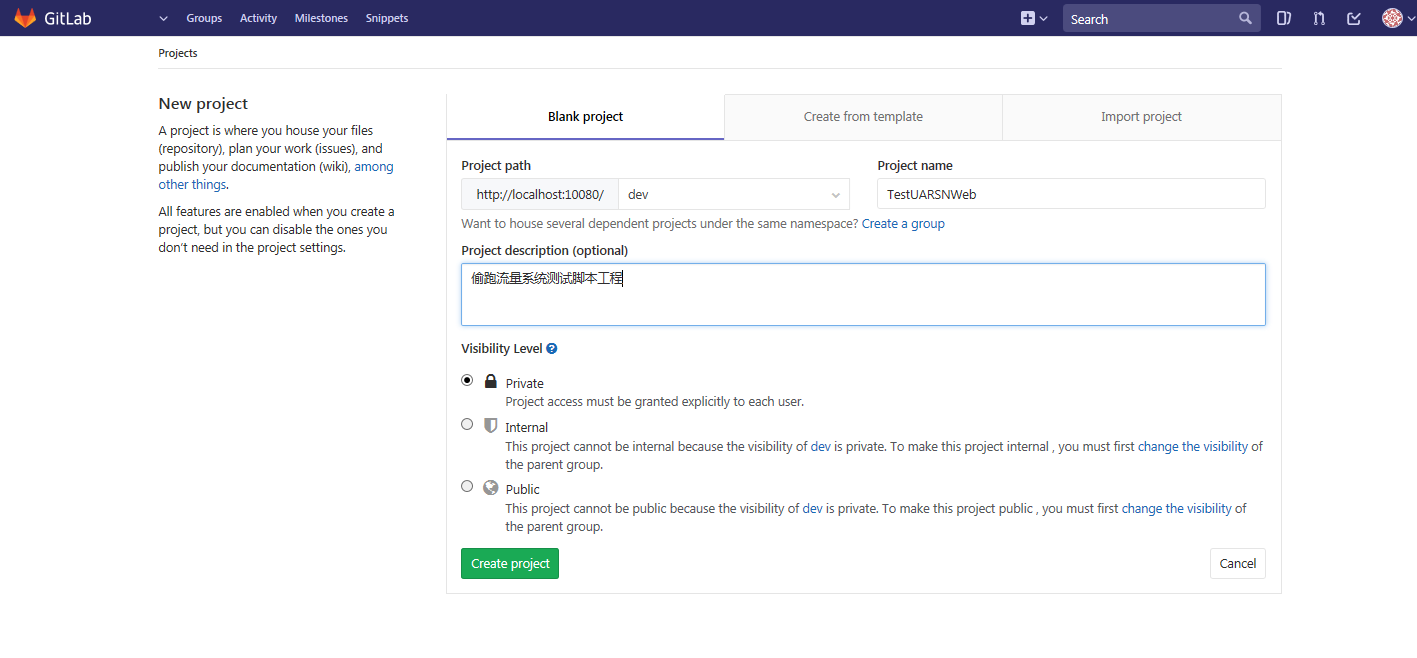


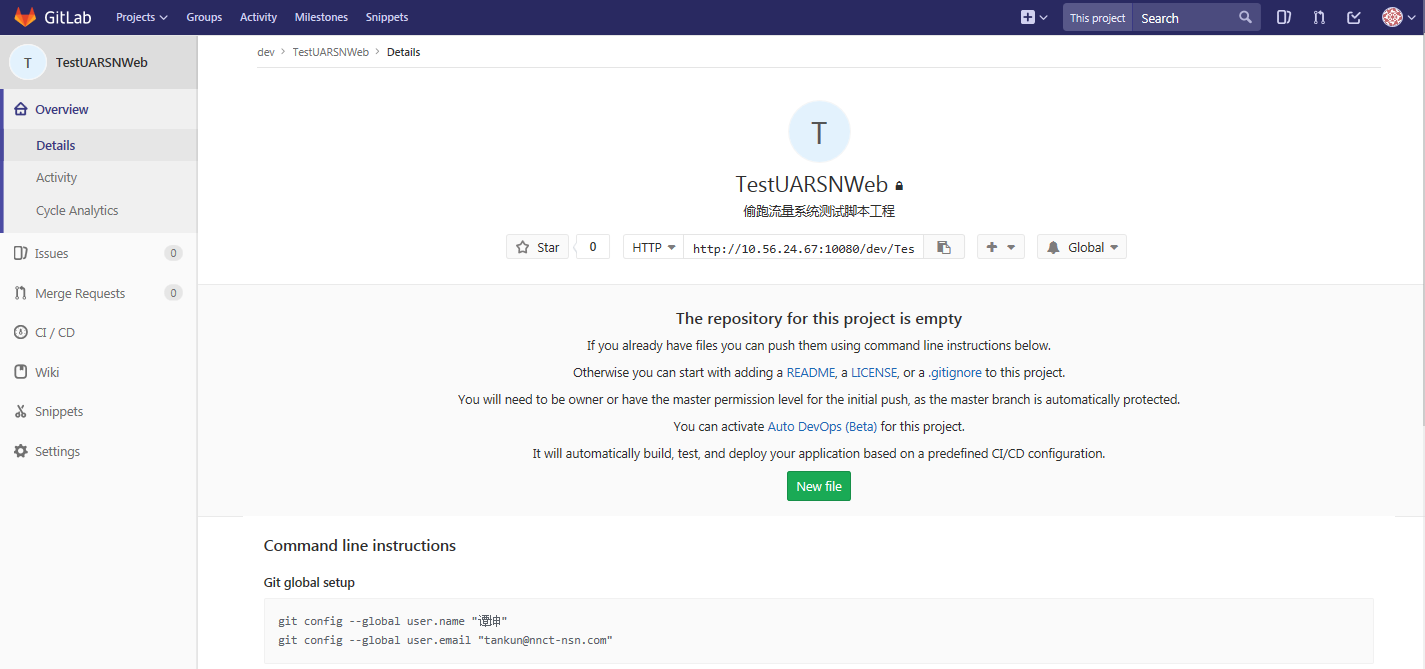
点击左侧菜单栏中的Overview，返回到组详细页面，点击此页面中的New Project，可以创建此组相关联的工程：





输入Project name, Visibility Level选择Private私有的【即只有这个组的成员可以看到此工程】，点击Create project:



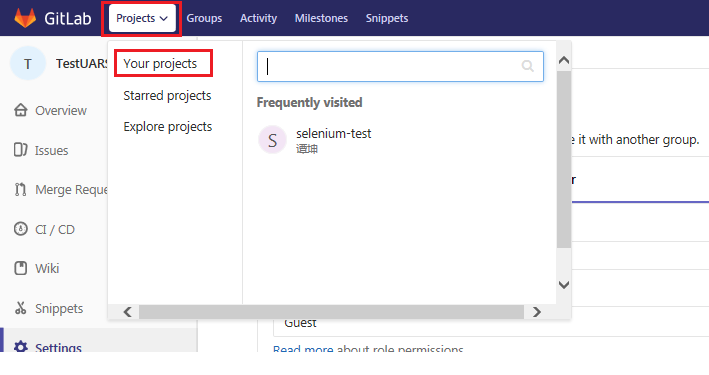


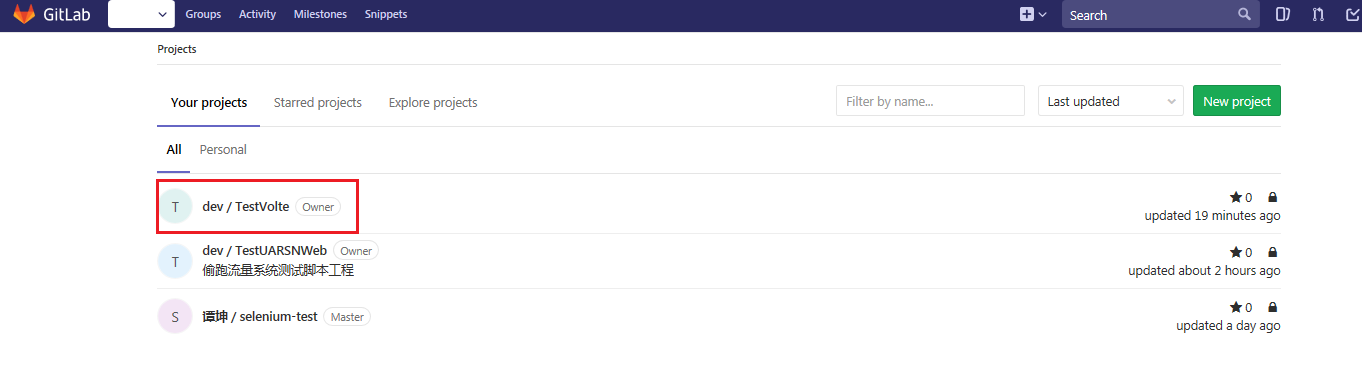
如上就是工程创建的相关操作；

## 上传本地代码

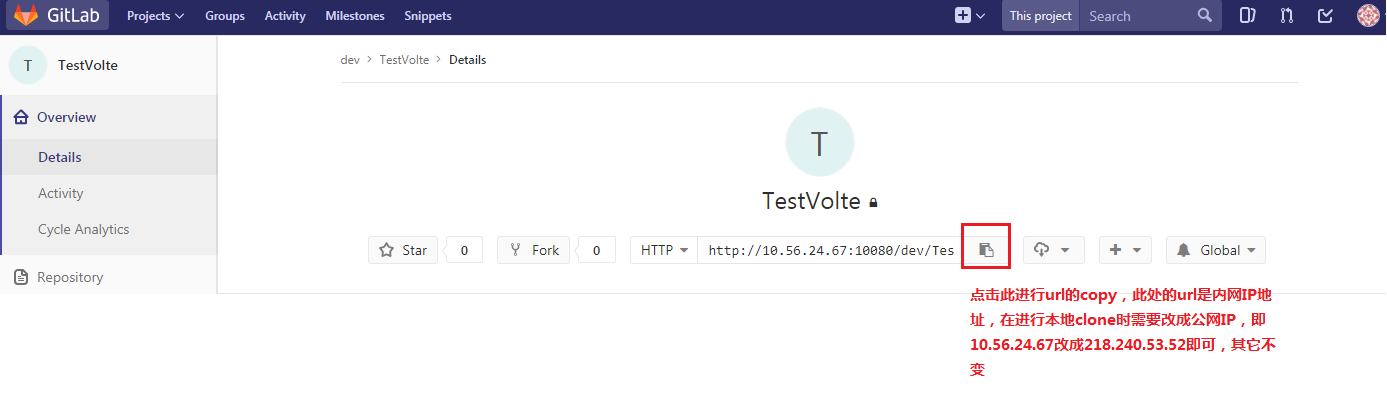
为了操作简单，借助于GitLab的辅助工具，即图形化界面：

GitLab中copy一下工程url，点击Project→Your projects：

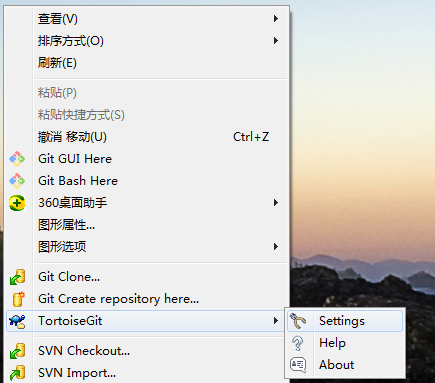




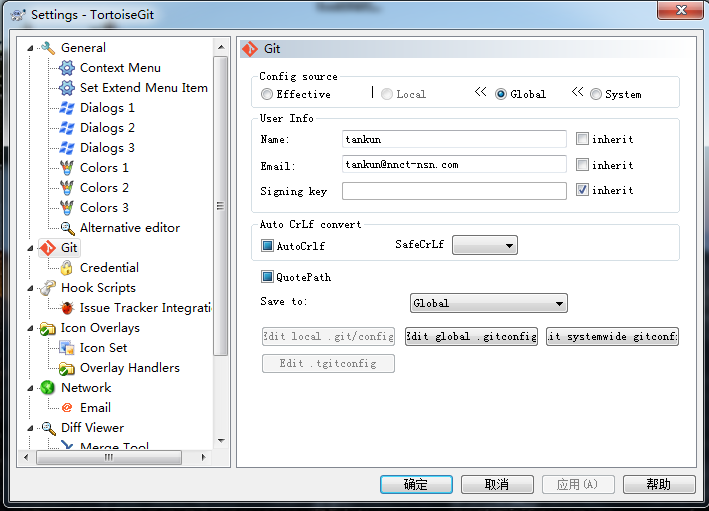
点击你需要的工程，点击途中红色框，即可以copy出url：



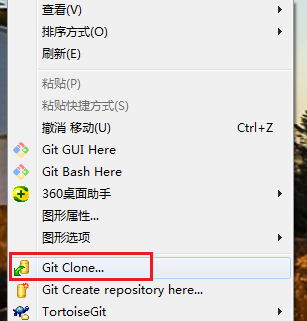
右键电脑桌面，点击TortoiseGit→Settings

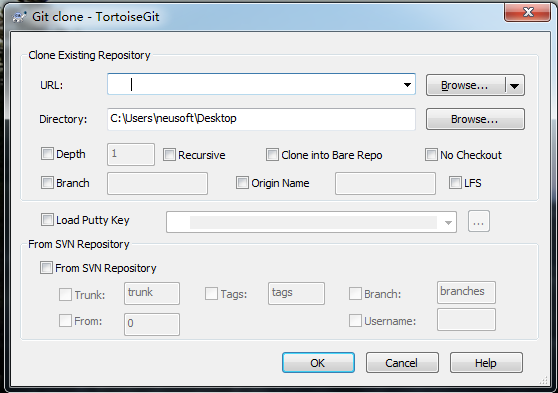


再点击Git→Global，去掉Name和Email后面inherit的勾选，输入您GitLab的用户名和Email，点击应用→确定：

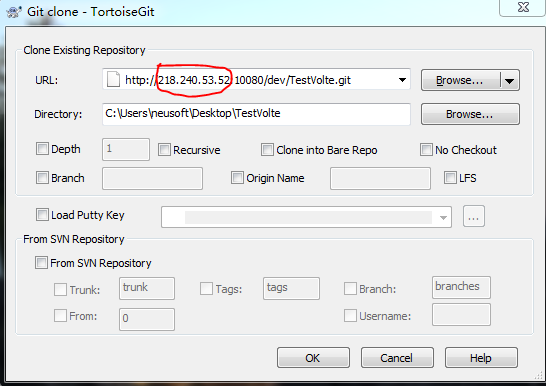


右键电脑桌面，点击Git Clone【即下载工程到本地】

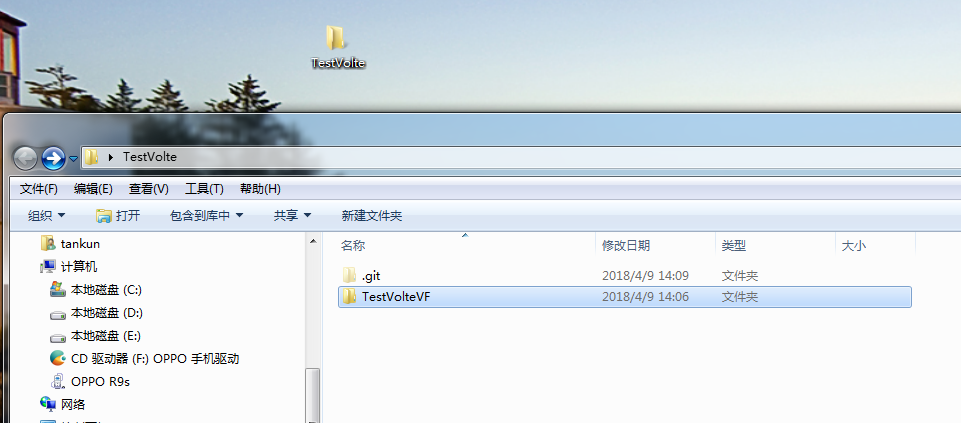




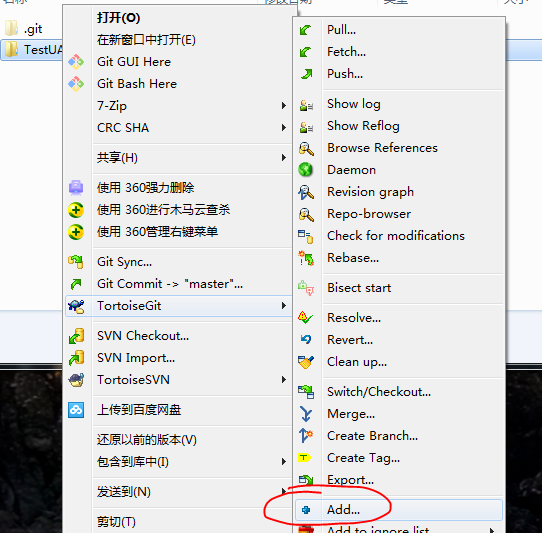
输入之前copy的urld到URL中，将10.56.24.67改成218.240.53.52，本地目录Directory我选择了桌面，大家可以自行选择，点击OK：

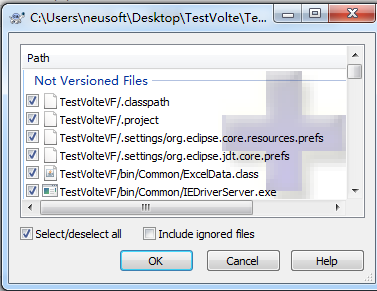


将本地代码文件放到clone下来的目录下：



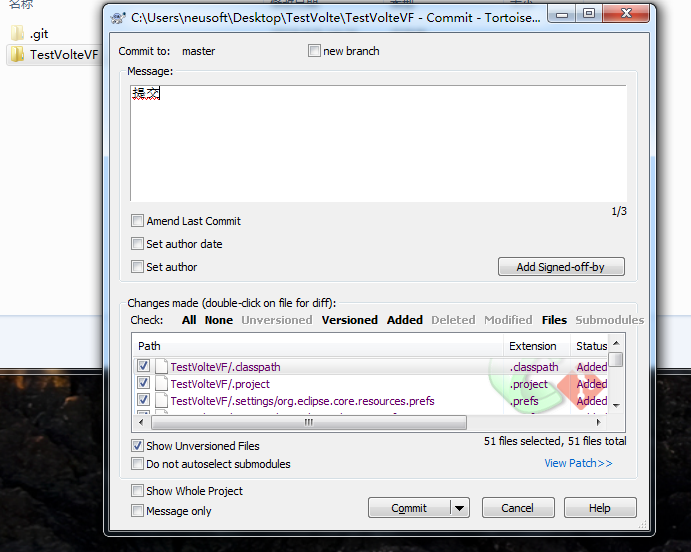
右键TestVolteVF，点击Add

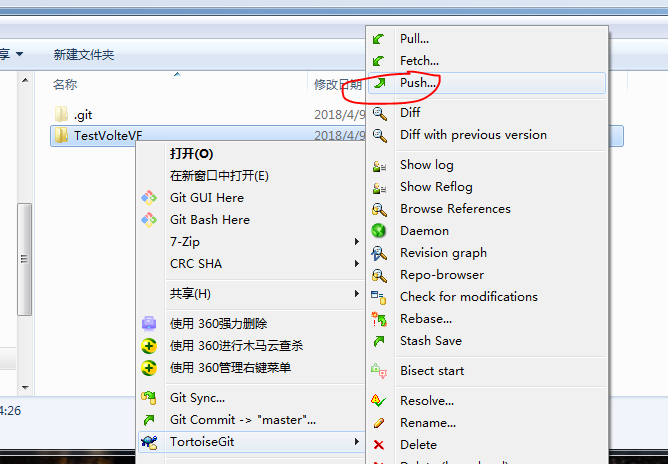


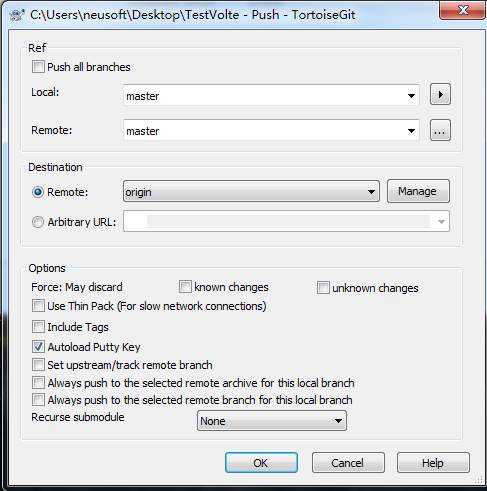


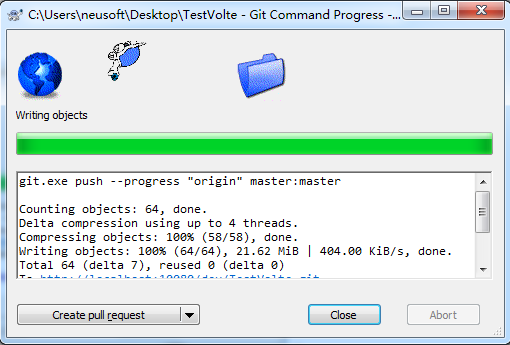
上传提交到master分支，只有工程owner和master人员可以上传到受保护的分支：

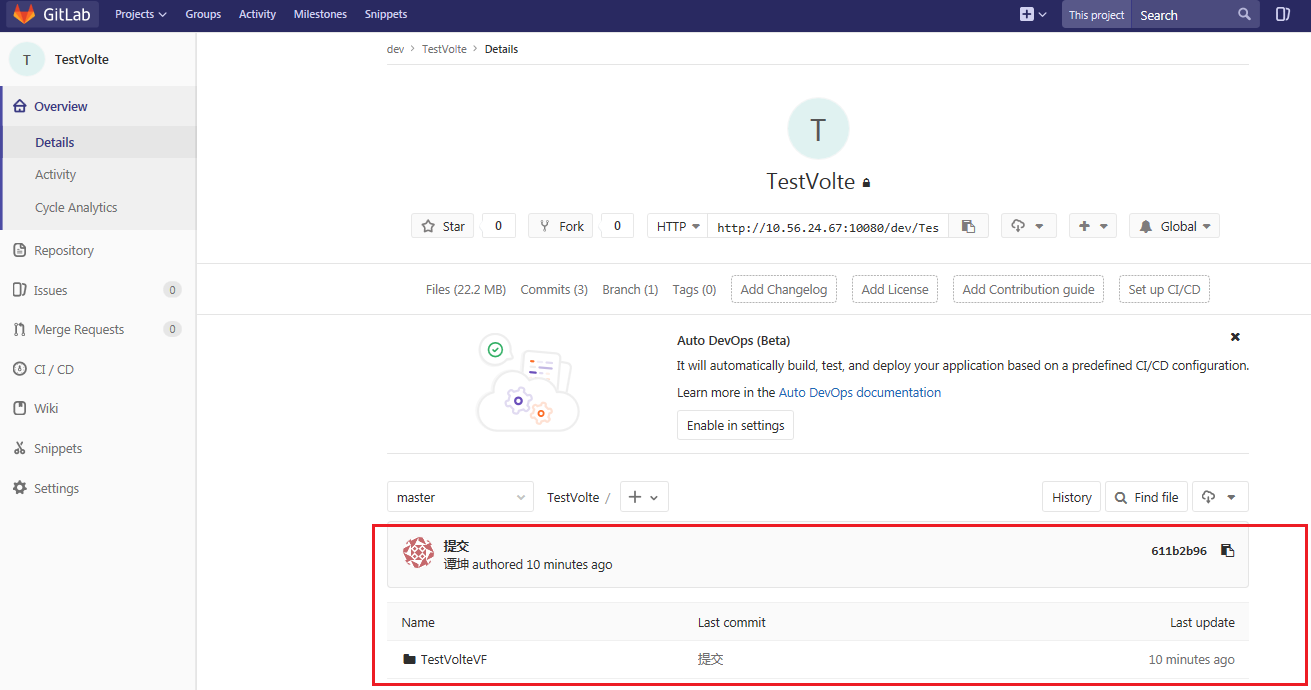












到此工程代码已经上传到GitLab中;

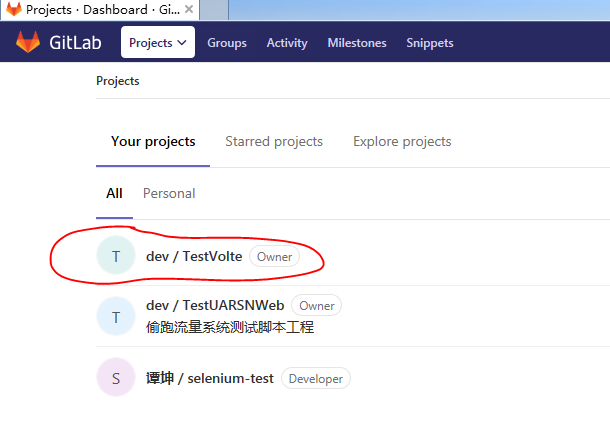
## Developer提交代码

由于Developer不能push代码到受保护的分支，如master分支，所以可以将代码提交到非保护的分支，提交完后在GitLab系统网站对提交的代码进行merge request，请代码有master权限创建人来决定是否可以或者拒绝；

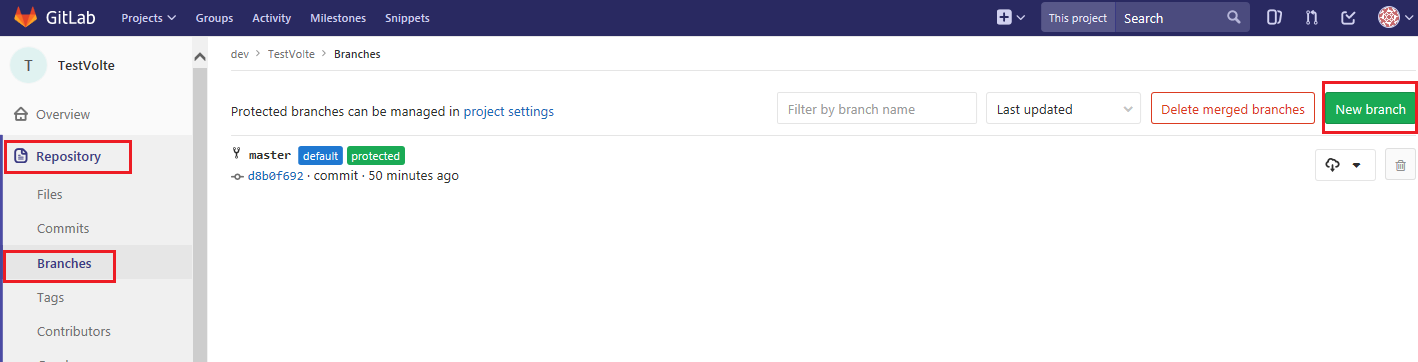
前提条件：

需要master权限者先创建同一项目的分支如branch1，则此分支就是非保护的分支；

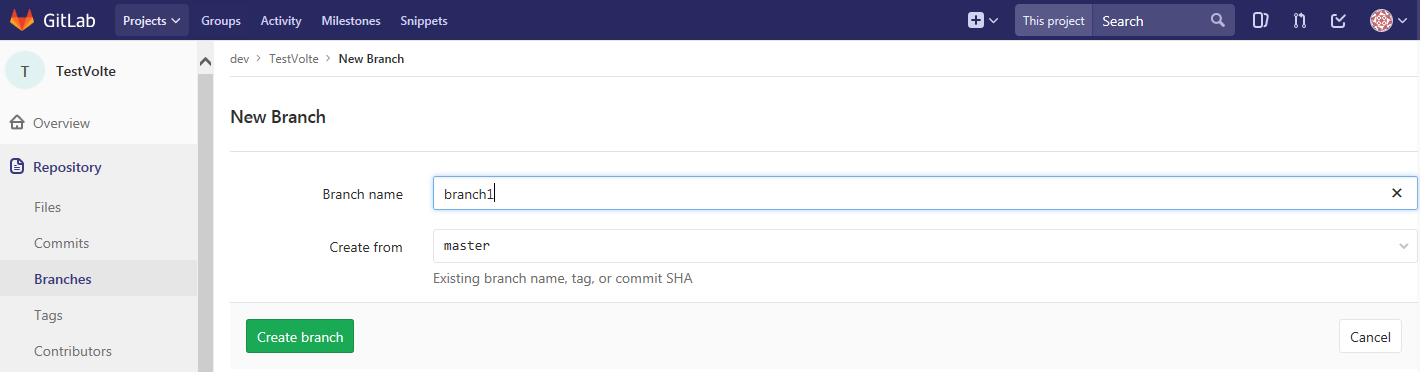
登陆GitLab系统，点击TestVolte工程



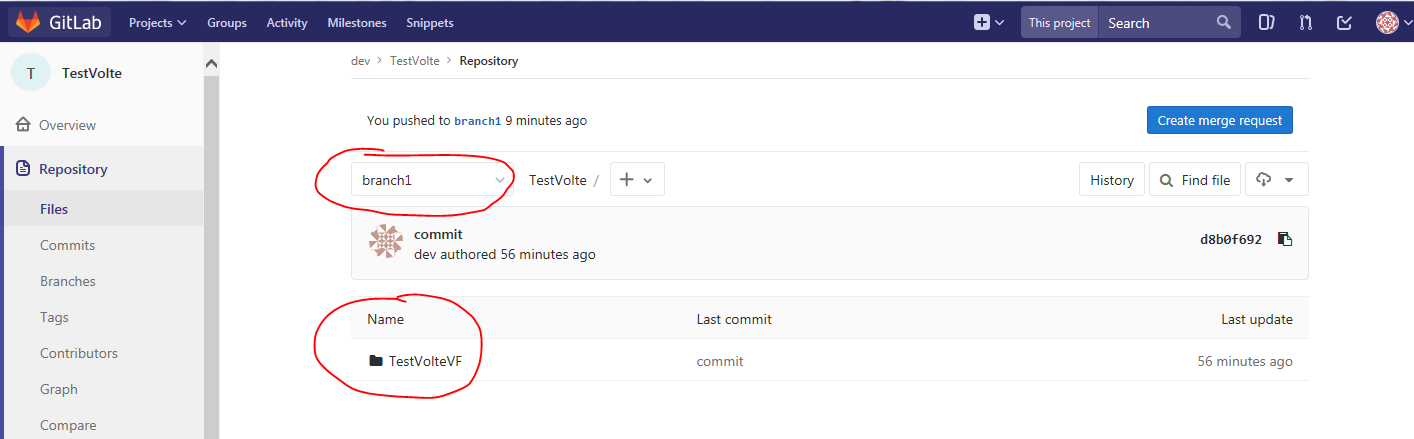
点击Branches→New Branch



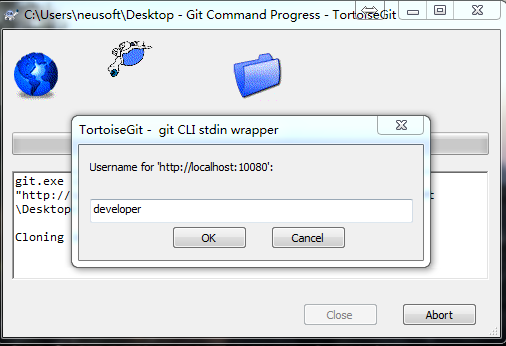
输入非保护分支的name，如下branch1，再点击Create branch

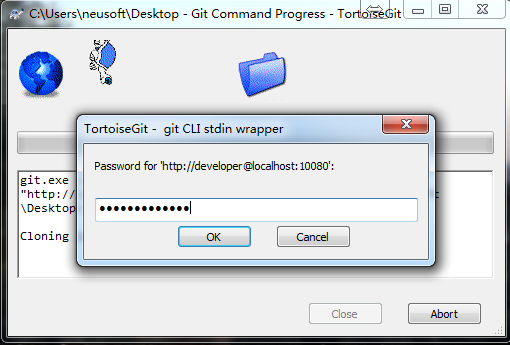


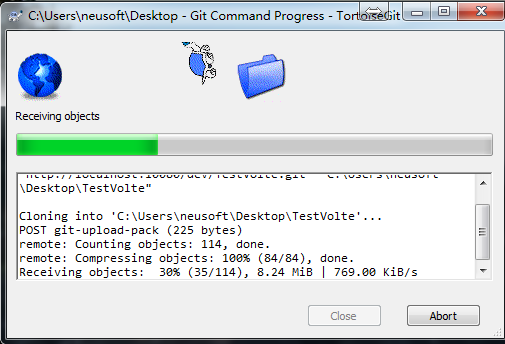
如下所示，同一个工程的分支就创建好了



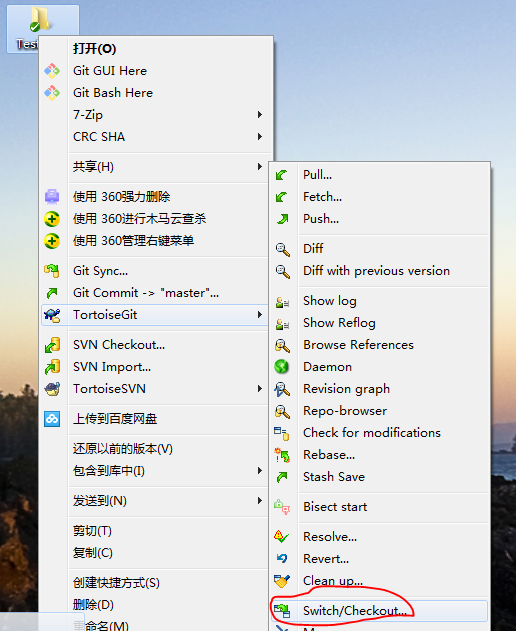
先将要修改的工程clone一下，即下载到本地：



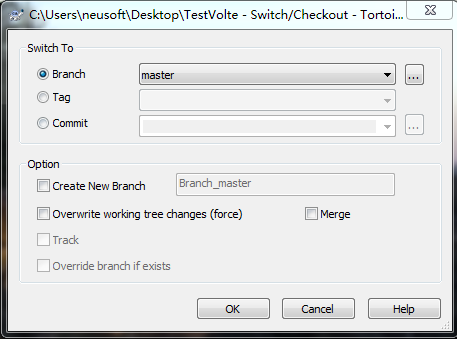




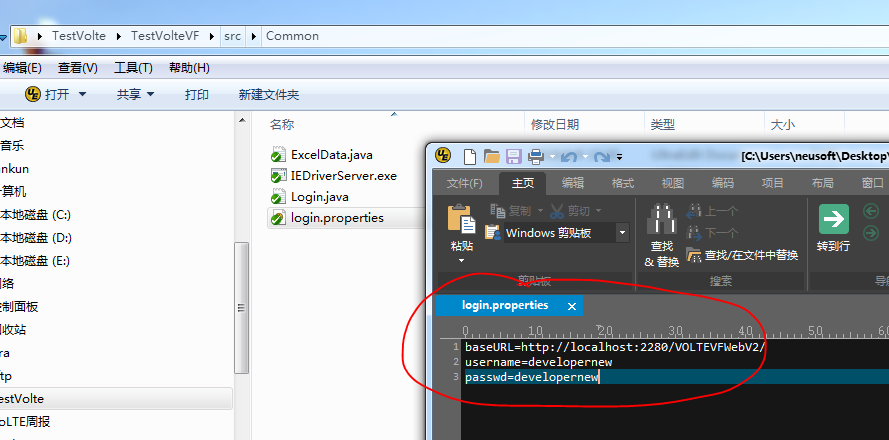
右键桌面刚才clone下来的工程，点击TortoiseGit→Switch/Checkout…



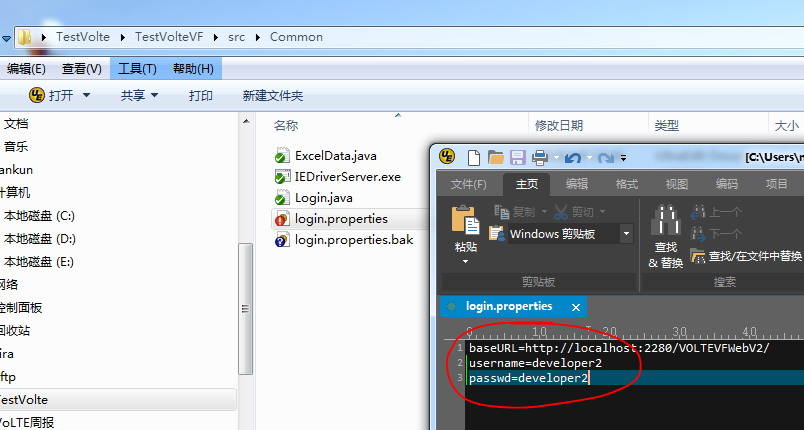
出现如下页面，点击OK，先使用master分支：



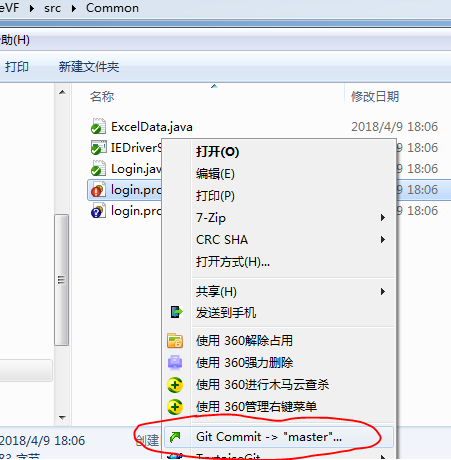
打开工程，修改其中的文件，如login.properties文件：



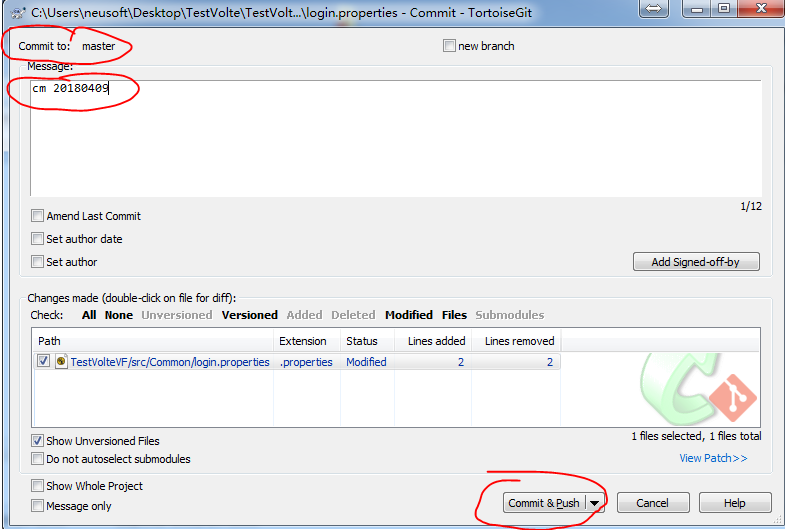
修改后如下：



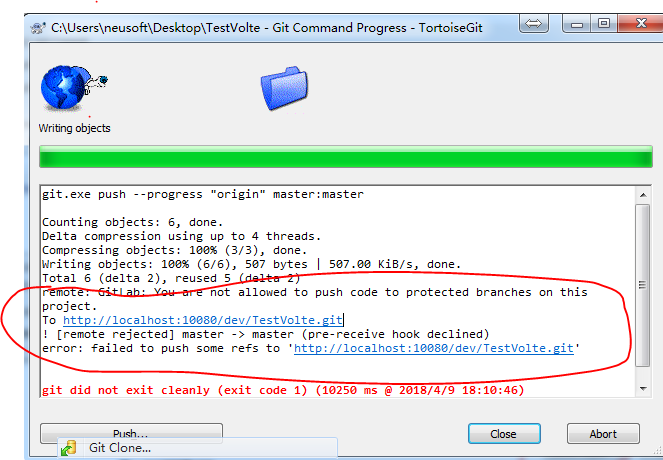
修改完后，进行commit&push【commit是本地提交，push是提交到远程服务器上】，右键修改的文件，点击Git Commit –＞“master”…：

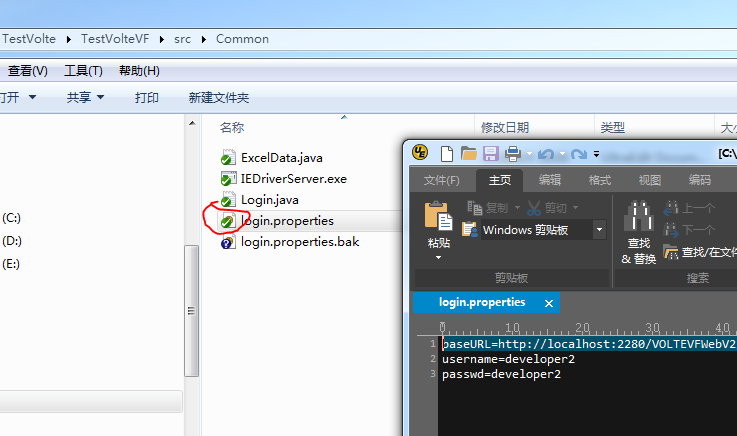


输入message,点击Commit&Push：



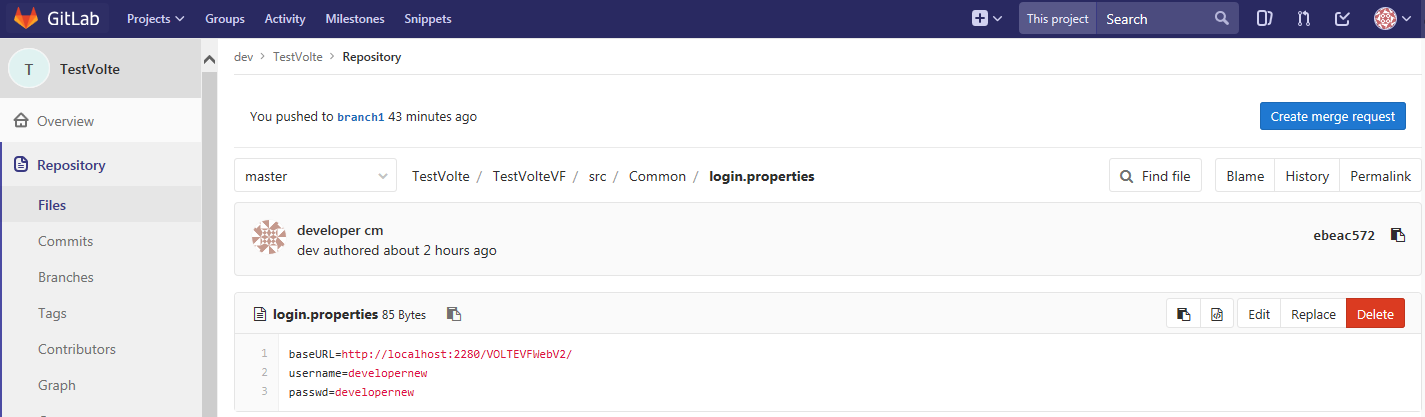
此时有如下报错，即不可以push【因为developer不能提交到远程服务器保护的分支master】，但是此时本地的commit已经完成：



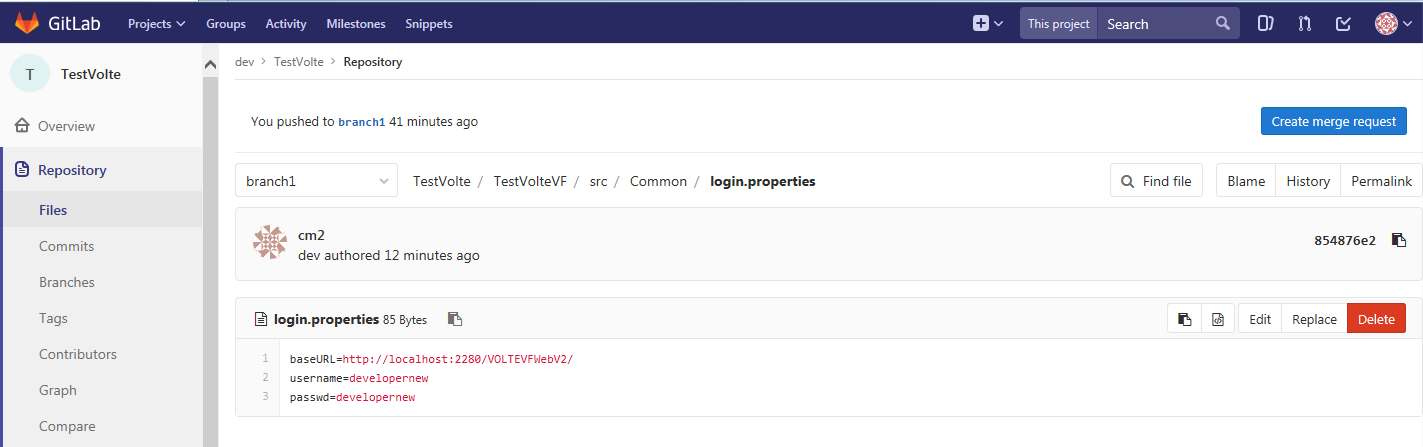


同时看看远程保护的master分支和非保护的branch1分支中此文件的内容：

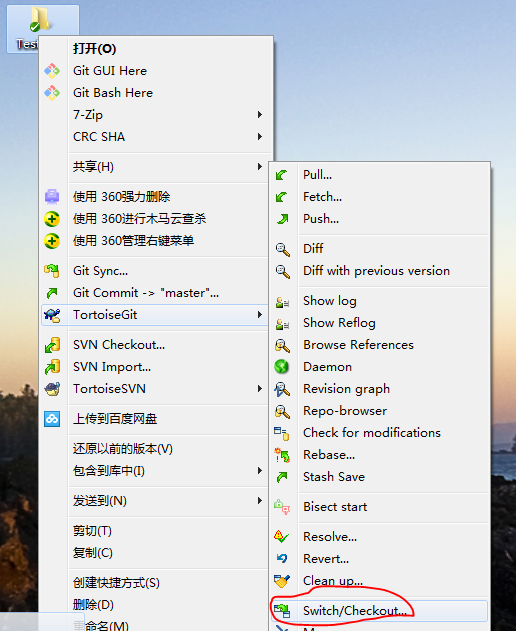
Master分支内容不变：



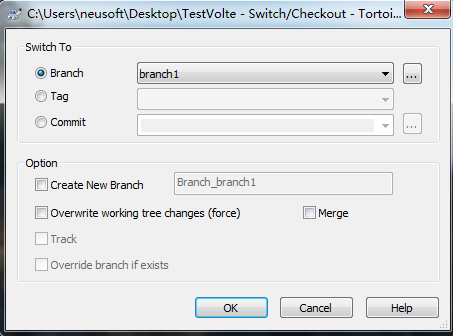
Branch1分支内容不变：



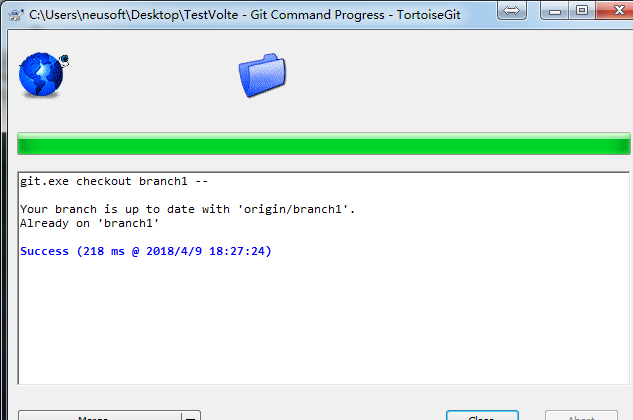
右键桌面修改完的工程，点击TortoiseGit→Switch/Checkout…



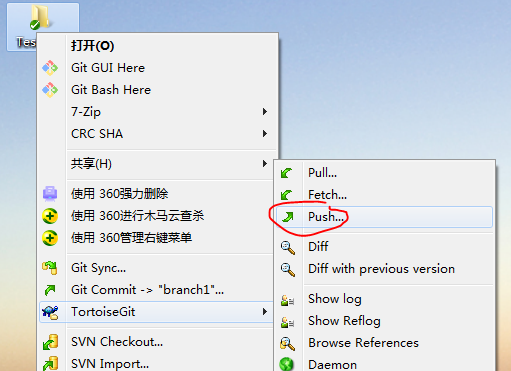
出现如下页面，Branch选择branch1，点击OK：



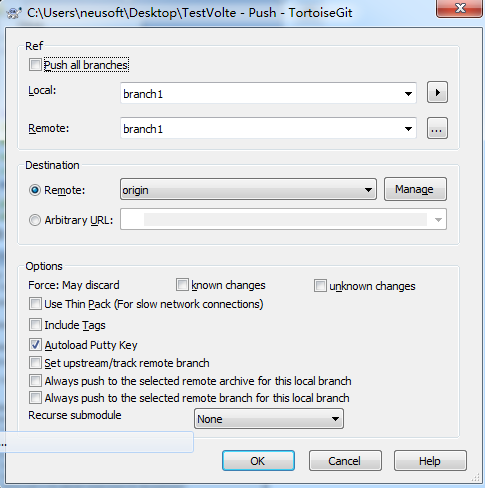
切换到branch1分支：



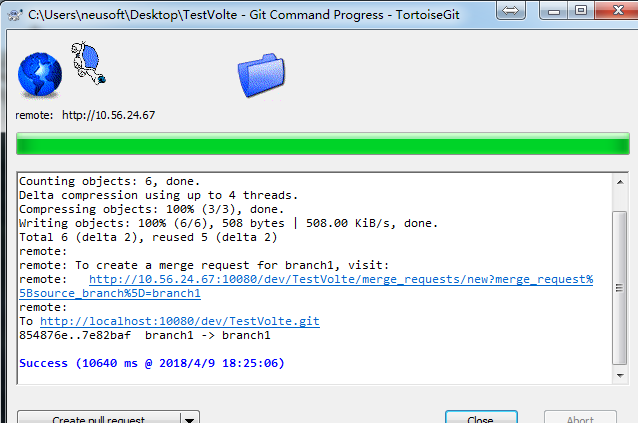
右键桌面修改完的工程，点击TortoiseGit→Push



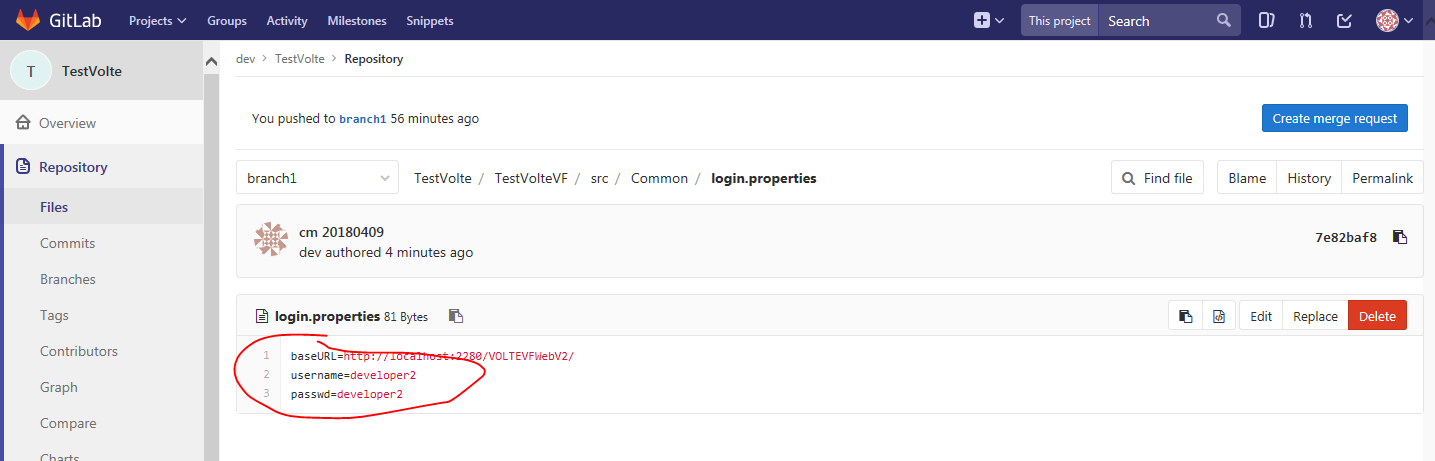
点击OK进行提交

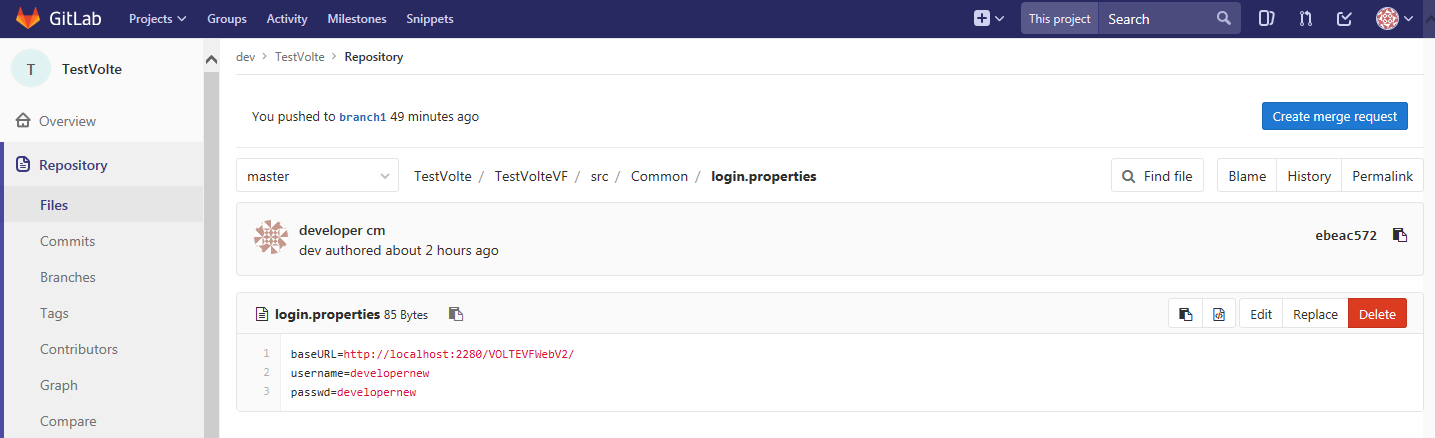


成功提交到远程branch1分支上：

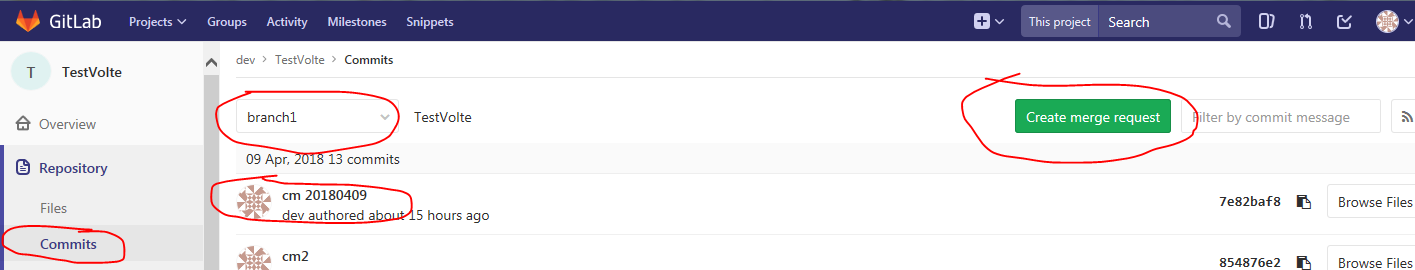


查看GitLab中branch1下文件是否改变【已改变】，但是master下文件没有改变：

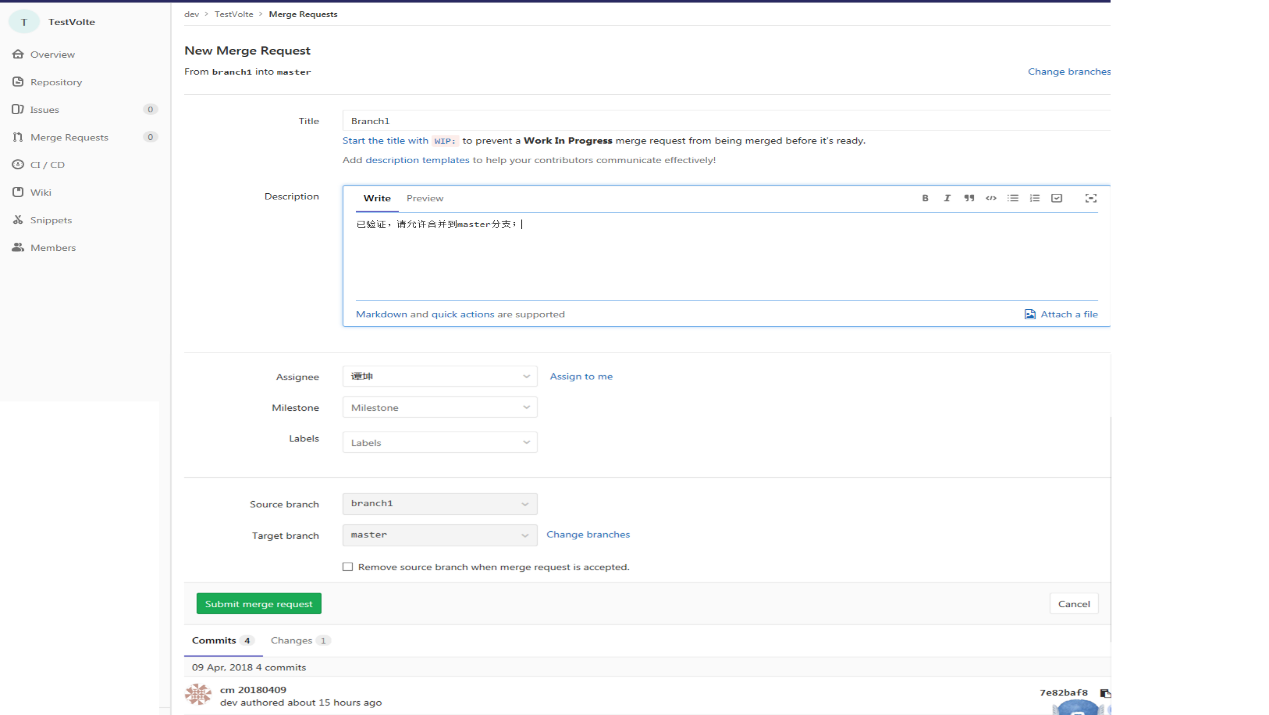




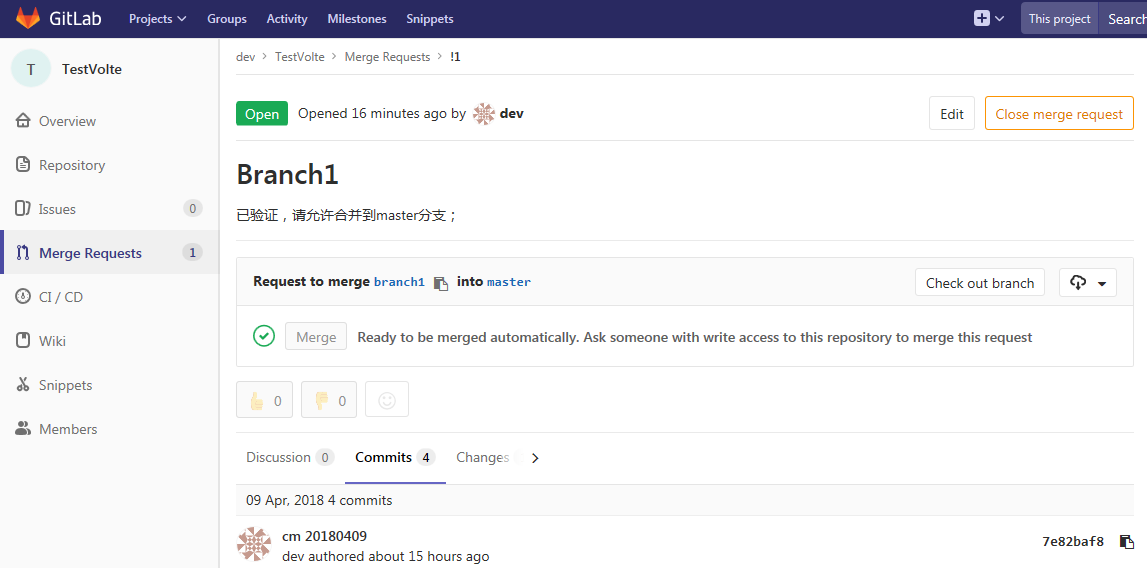
此时要想让master分支的此文件变更，只能通过向master权限者进行merge request，申请合并此文件到master分支：



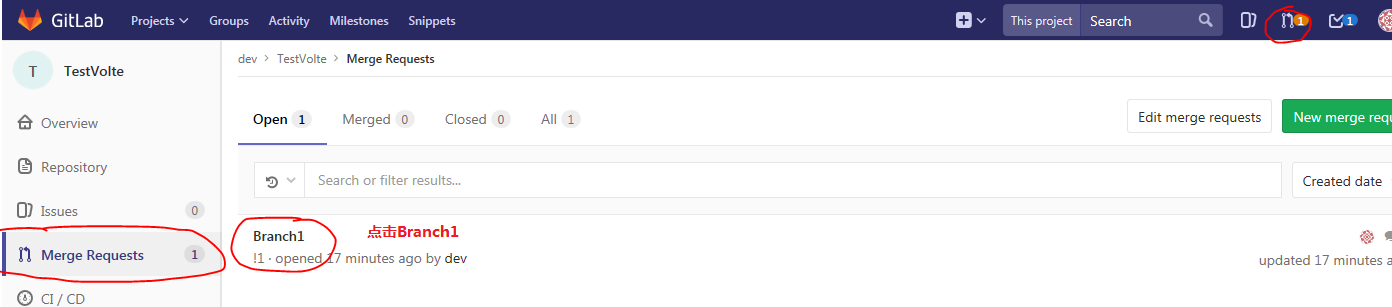
输入相关信息，Assignee里选择master权限者的，指派给他，请他允许merge，天机Submit merge request



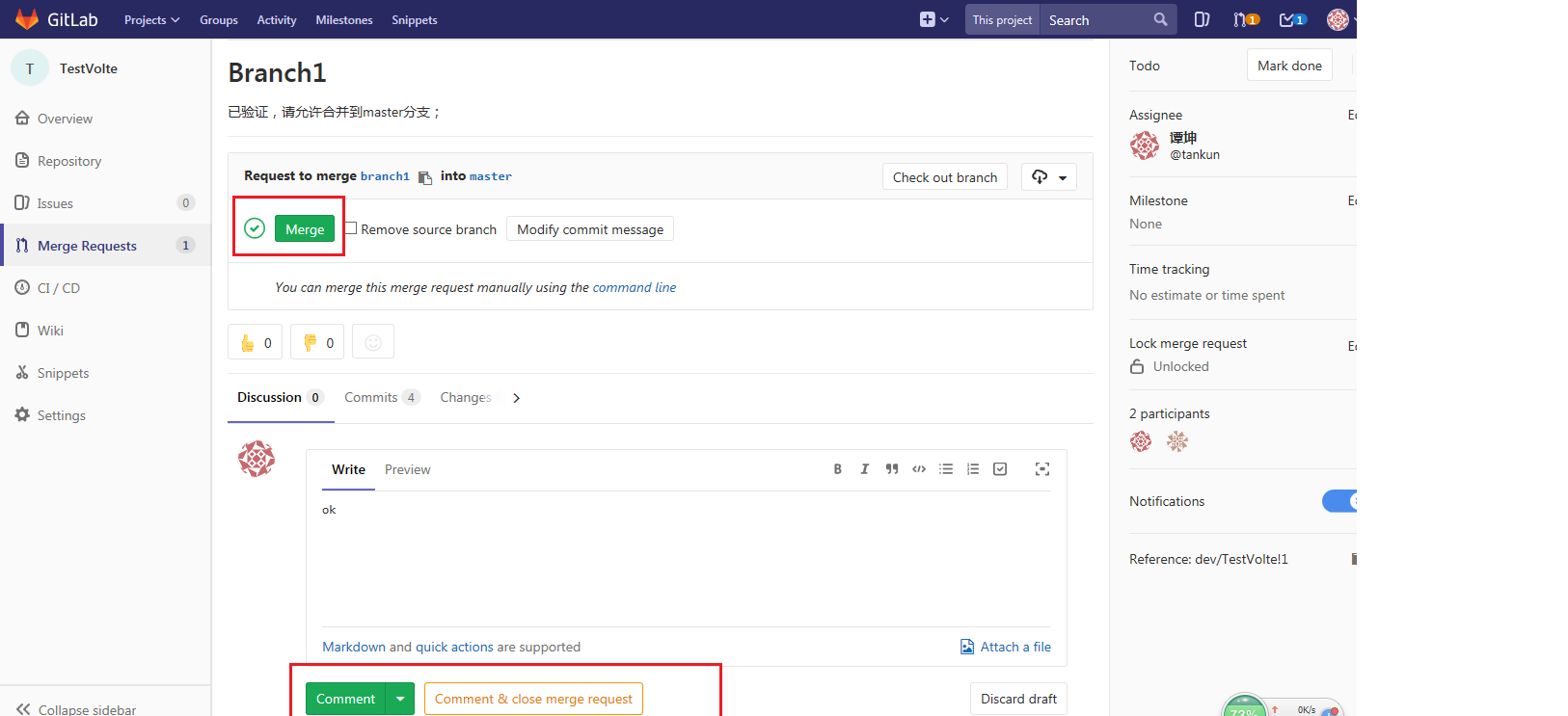
此时你的GitLab下就出现一个Merge Requests记录，等待master允许merge：

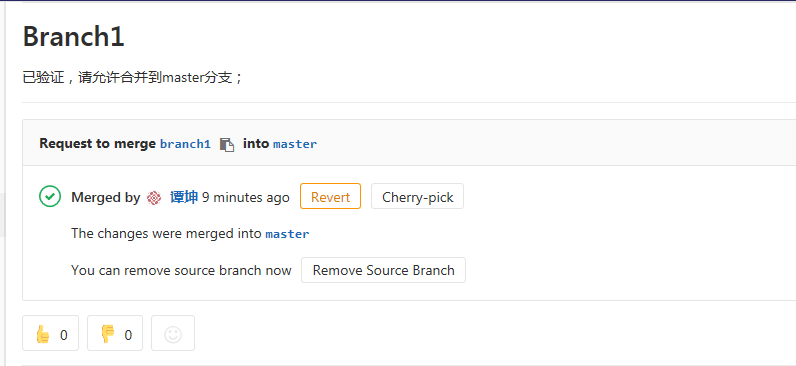


进入master的GitLab：

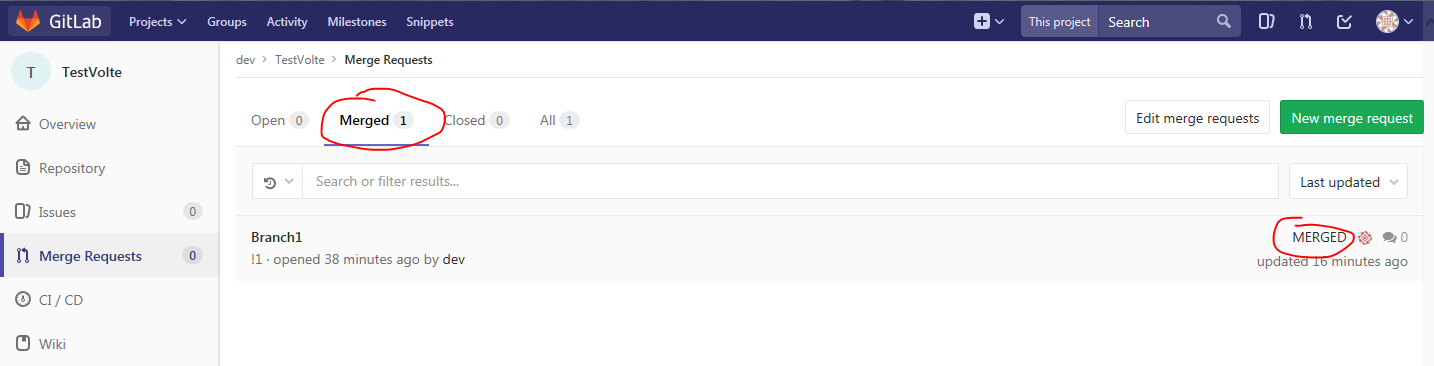


点击Merge则允许，master分支中此文件就会改变

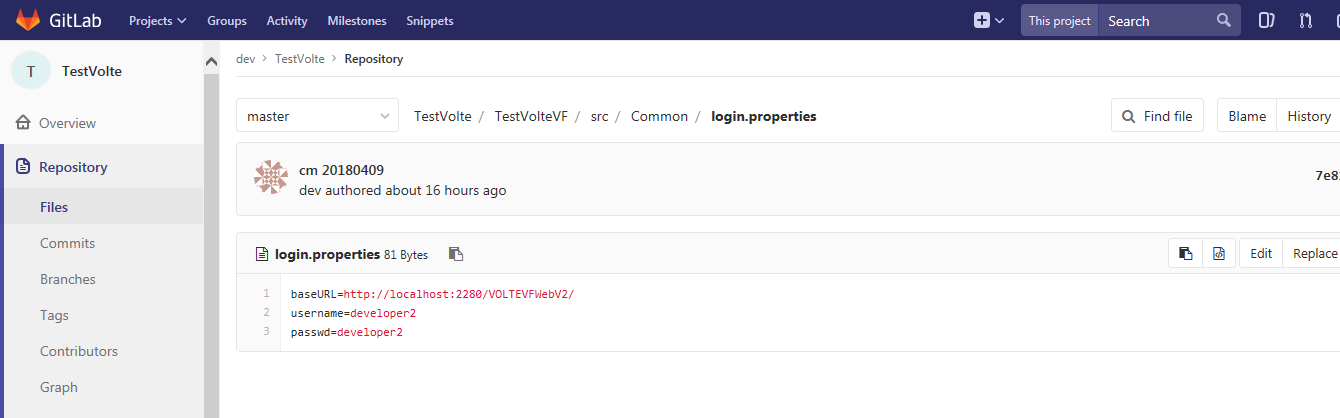




进入提交merge请求用户的GitLab，可以看到没有Merge Requests，只有Merged的记录：

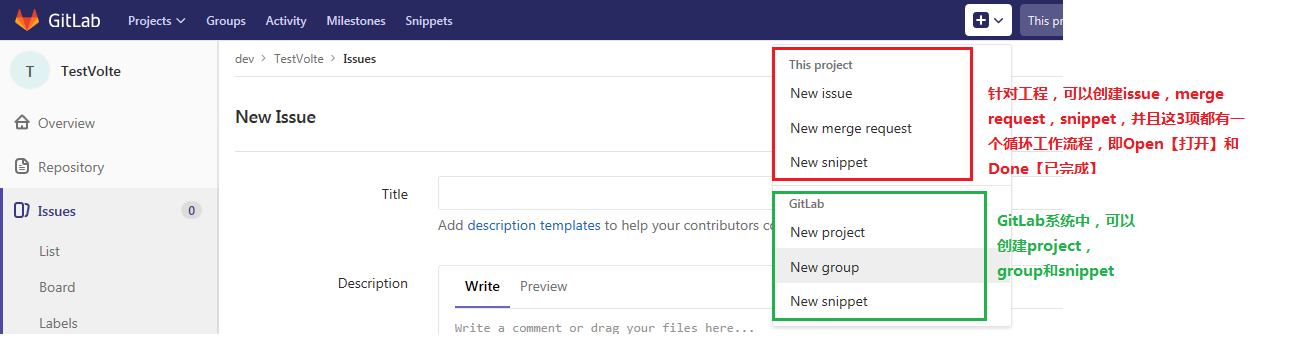


查看master分支上文件内容，已经变更为最新的：



## 其它：

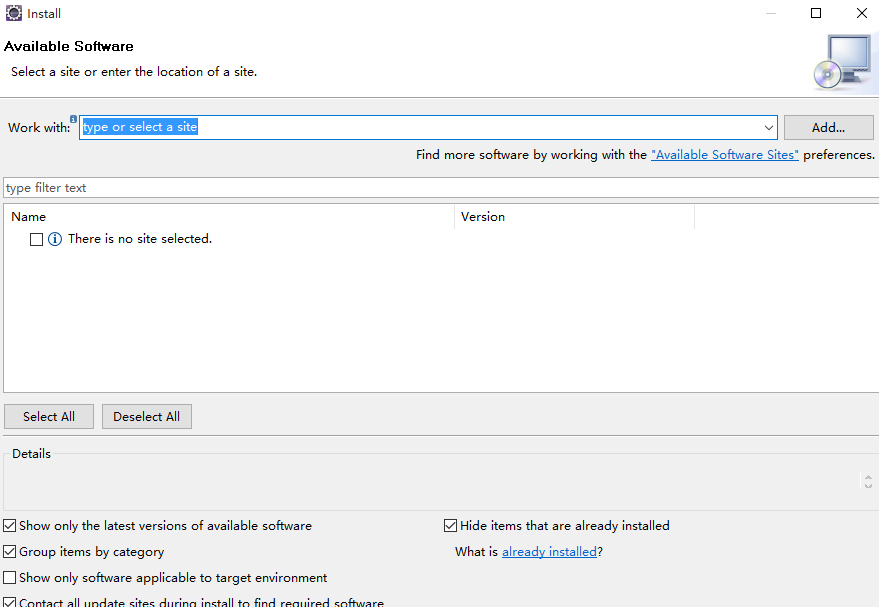
GitLab中还可以提出问题，指派给能解决此问题的人员，整个问题提出和解决是个循环工作流，具体操作不讲述了，各位看看系统，点击操作一下就能学会使用：



# Eclipse中使用Git说明

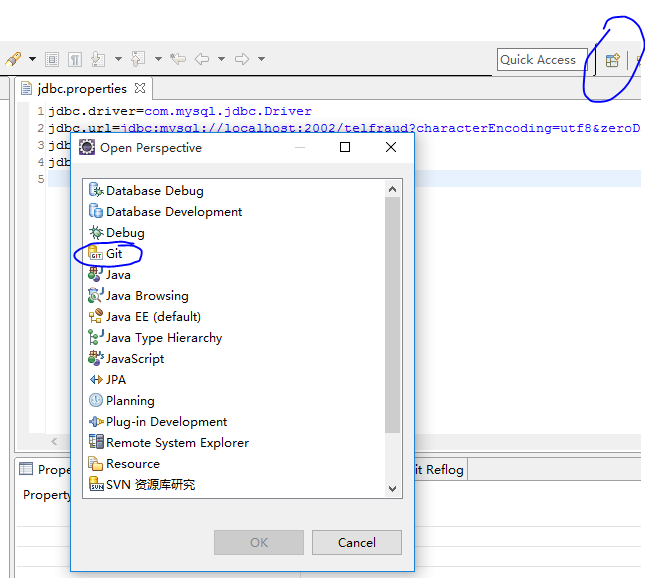
## 安装Git插件

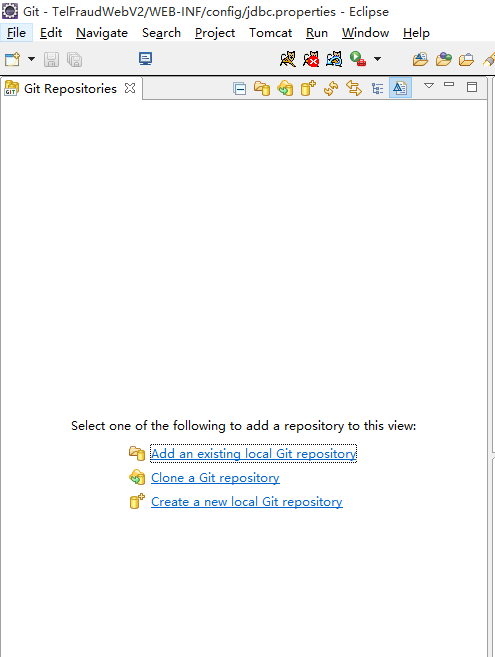
打开eclipse，点击Help→Install new Software…



点击Add，输入EGit和<http://download.eclipse.org/egit/updates/>，确定，默认步骤执行下去；

Eclipse中就多出Git插件：如下：





## 配置eclipse中的Git

1. **Windwos-->Preferences-->Team-->Git-->Committing**

把User Staging....前面的钩去掉， 否则在后面提交时会报"There are no staged files" 错误

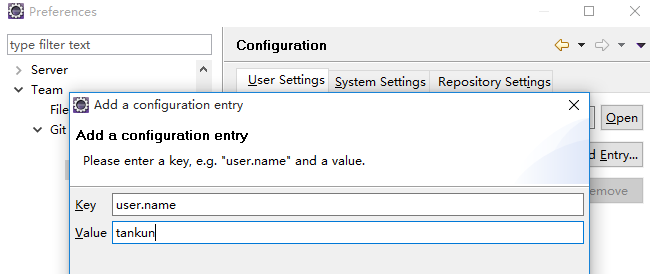
1. **对Git插件进行配置[用户名]和[邮箱]**

Windwos-->Preferences-->Team-->Git-->Configuration-->Add Entry

分别配置 user.name和user.email

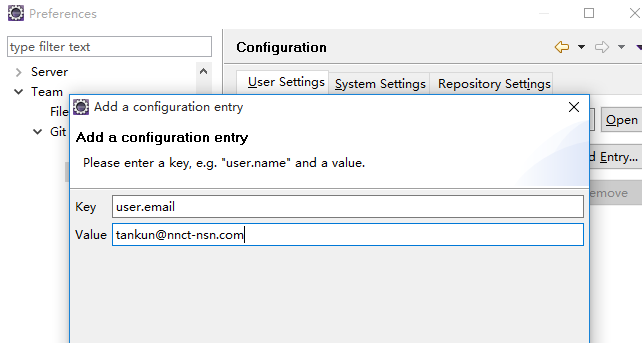
如：Key→user.name

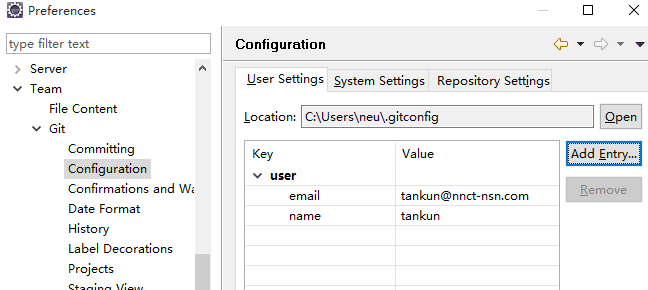
Value→tankun



Key→user.name

Value→tankun@nnct-nsn.com





## 发布一个项目

1. **选中项目右键-->team-->Share Project...-->Git**

在User or create respository...前打钩(创建一个Git仓库,仓库位置可自定,一般在项目下)

Create respository-->Finish

1. **完成上一步骤后会发现,项目名加了NO-HEAD,这表示该项目已有了仓库但未提交**

右击项目-->team-->commit-->随便填写些文字(表修改内容)-->

下面选择要提交的内容commit或commitandpush(会有个远程git仓库,如果已有，可提交到该仓库)

提交完成会看到项目名加了master(仓库中一个默认分支)

## 检出项目到本地

Import-->Project from Git-->Clone URI-->填写码云上项目路径-->选择检出路径

选择Import existing Eclipse projects-->Finish 完成