恒大智慧小区项目

**GIT 用户手册**

**项目团队名称**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
|  |  | | | |
|  |  | | | |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | | | |
|  |  | | | |
|  |  | | | |
|  | | | | |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |

目 录

1 安装 Git 客户端工具 5

2 配置 Git 6

2.1 生成 SSH Key 6

2.2 导入 SSH 公钥 7

2.3 初始化Git 9

3 Git CMD 命令 10

3.1 Git Clone 仓库 10

3.2 Git 本地仓库结构 12

3.3 Git add 命令 12

3.4 Git rm 命令 14

3.5 Git commit 命令 15

3.6 Git push 命令 16

3.7 Git pull 命令 17

3.8 Git branch 命令 18

3.9 Git merge 命令 21

3.10 Git fetch 命令 25

3.11 Git tag 命令 25

3.12 Git 替换本地改动 26

3.13 Git 父项目和 fork 项目间的合并 27

3.13.1 合并其他子项目 28

3.13.2 合并父项目代码到子项目 30

3.13.3 合并子项目代码到父项目 30

4 Git Web 界面 32

4.1 添加用户 32

5 Git GUI 35

# 安装 Git 客户端工具

市面上有很多Git 的客户端工具，有命令行的，也有图形界面的，开发人员可以根据自己的喜好，自己去下载安装这些工具，本文档只介绍其中一种。

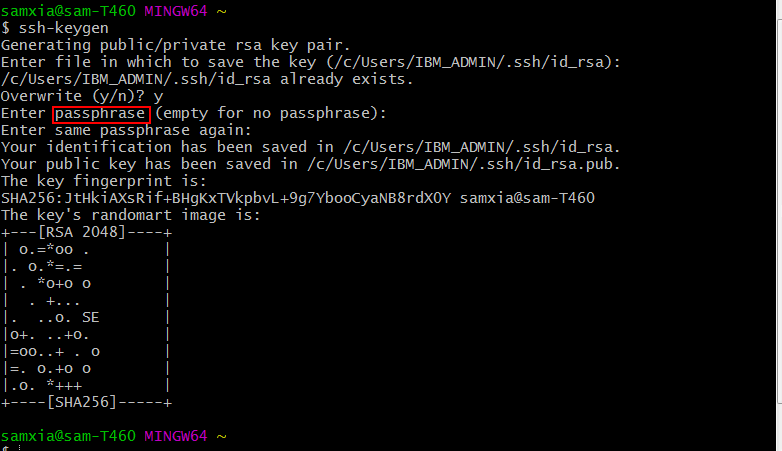
官方下载地址： <https://git-scm.com/downloads>

下载完成后，双击该 exe 文件开始安装，修改安装路径后，可以一直点击 “Next” 按钮直到安装完成。

# 配置 Git

从Git 仓库里 push 或者 pull 代码时，需要输入用户名和密码，我们也可以在开发机器上生成 ssh-key, 然后在 Git 服务器上把这个 key 加到自己的 profile 里，使用 ssh 方式连接 Git 仓库，下面详细介绍具体步骤。

## 生成 SSH Key

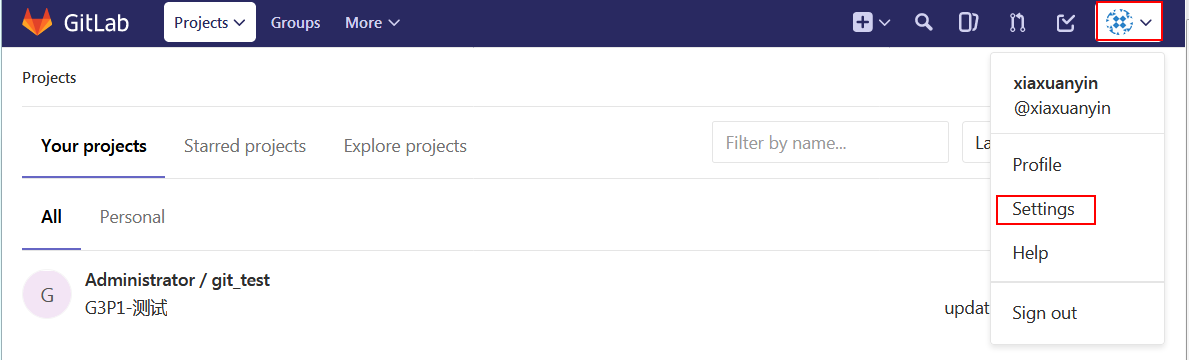
在Windows 开始菜单里找到 Git -> Git Bash 子菜单，运行它。在 Git Bash 窗口里，运行 ssh-keygen 命令，如下图。

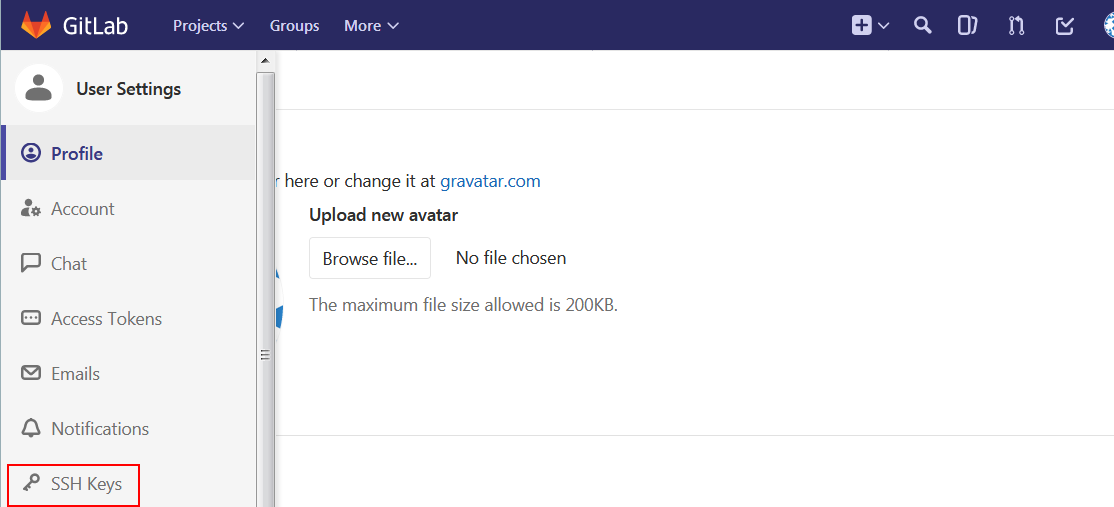
在提示 “Enter passphrase” 时，可以输入密码，或者回车，这样以后跟Git 仓库的连接不需要输入密码。

在本例子中，生成的ssh 公钥文件为 c:\Users\IBM\_ADMIn\.ssh\id\_rsa.pub， 用Notepad 打开该文件。

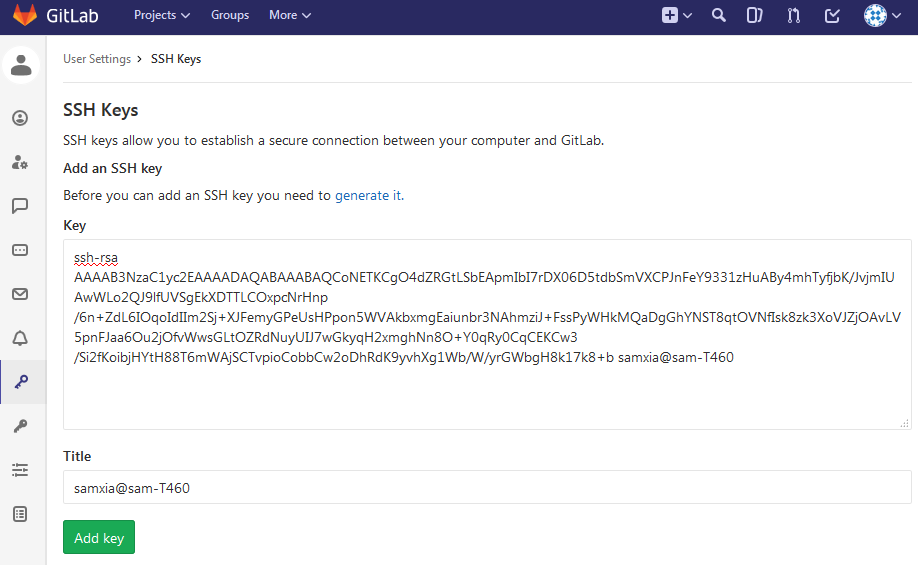
## 导入 SSH 公钥

访问Git 服务器 ，登陆成功后可以查看自己有权访问的所有仓库（Project），如下图。

点击右上角的图标，选择“Settings” 子菜单，如下图。

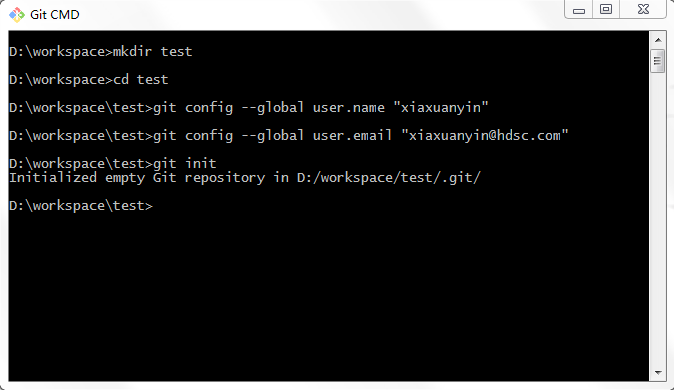
在用户配置页面，选择”SSH Keys” 如下图。

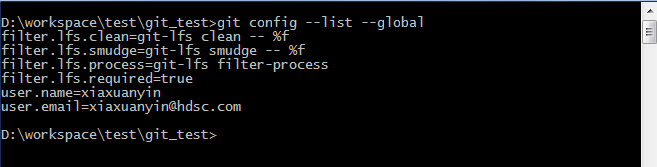
把写字板里id\_rsa.pub里的内容拷贝到Key 输入框里，如下图



点击 “Add key”按钮。

## 初始化Git

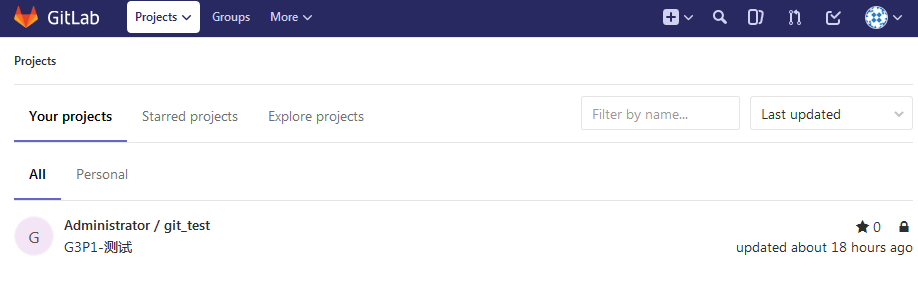
创建一个本地目录，比如 D:\workspace\test，运行如下命令配置一些全局变量并初始化本地仓库。

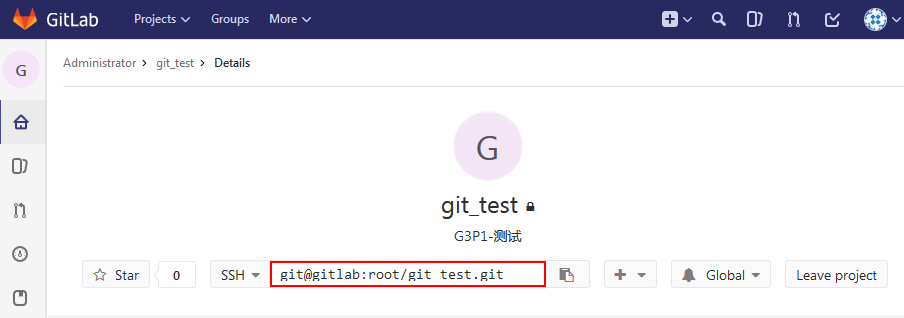
查看全局设定

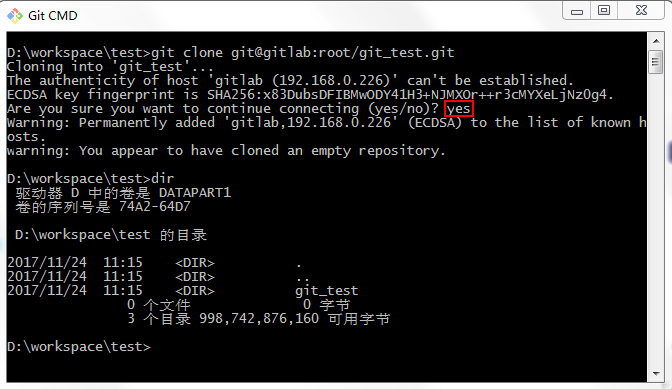
# Git CMD 命令

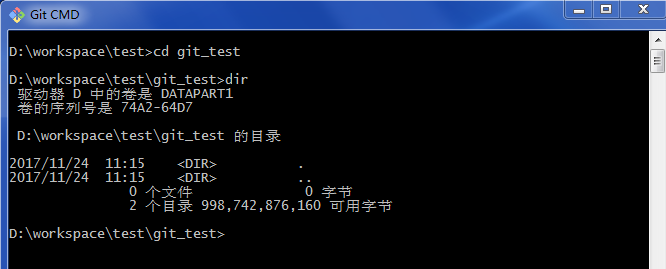
## Git Clone 仓库

登录到 后，可以看到自己能访问的项目如下图。

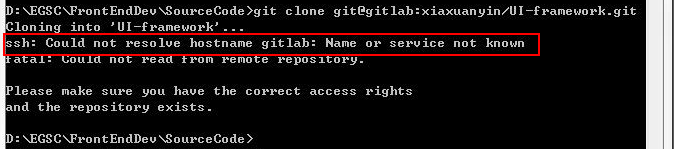


点击项目 git\_test，可以看到项目的基本信息，把项目仓库的 ssh 地址拷贝下来如下图，

在 Git CMD 或者 GitBash 窗口里，在刚创建的本地仓库 test 目录下，运行Git Clone 命令如下图



因为现在仓库里是空的，所以 clone 到本地的仓库没有任何文件。

如果 Git clone 出现如下错误，

原因是不能解析 gitlab 域名，解决方法有三，任选其一

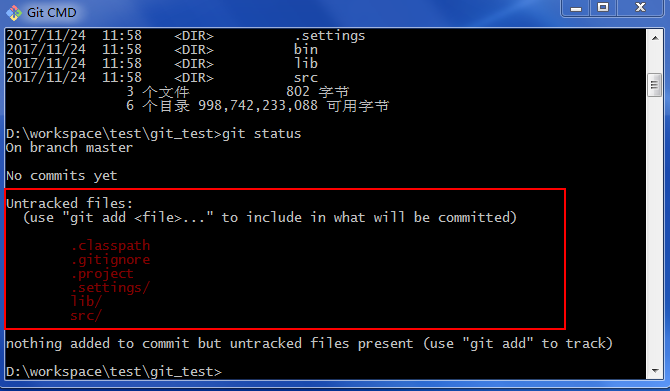
* 将工作机添加到 HDSC 域里，方法请百度。
* 将 hdsc.com 加到DNS Suffix 表里，方法见百度。
* 在 git clone 时添加域名，即把 gitlab 替换为 gitlab.hdsc.com

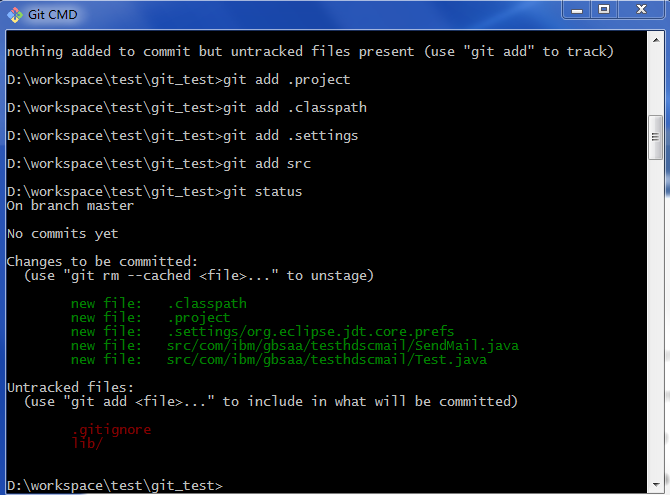
## Git 本地仓库结构

Git 本地仓库包括三部分（三棵树），一个是我们的工作目录，也就是我们通常在 Windows 里看到的目录结构；一个是 Index(Stage) 树，当我们在工作目录里添加了新的文件，修改了文件或者删除了文件，需要通过 Git add 把这些更新到 Index 树里，只有在 Index 树里的更新才能被 commit 到 Head 树里；最后一个是 Head 树，可以从 Head 树 push 更新到服务器或者从服务器 pull 更新到本地。

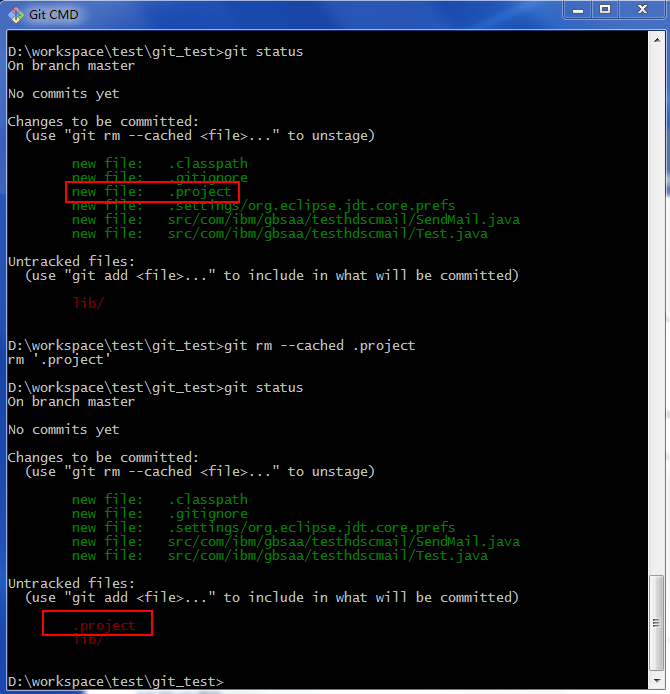
## Git add 命令

工作一段时间后，在 git\_test 目录下已经有了变动，需要提交。可以运行 git status 来查看当前代码



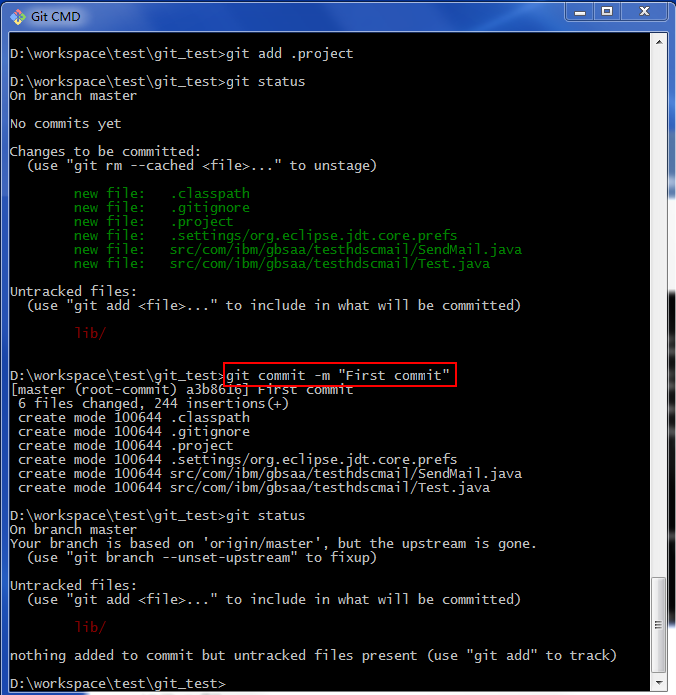
我们可以运行 git add把一个文件，一个目录或者多个文件加到index 树里，如下图。

## Git rm 命令

我们有时可能把一个还没完成的文件不小心更新到 index 树，如果这个文件被 commit 到服务器可能带来不良后果，我们可以运行 git rm 命令把该文件从 index 树里删除。

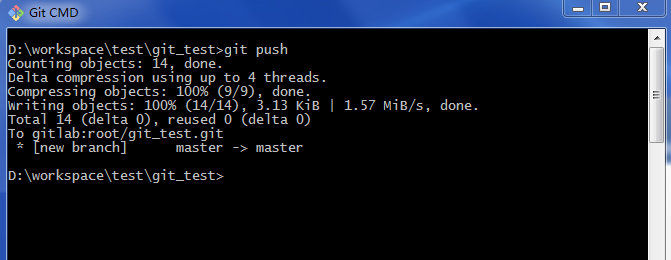
如果要从工作目录和 index 树里删除某文件，可以运行 git rm –f 命令。

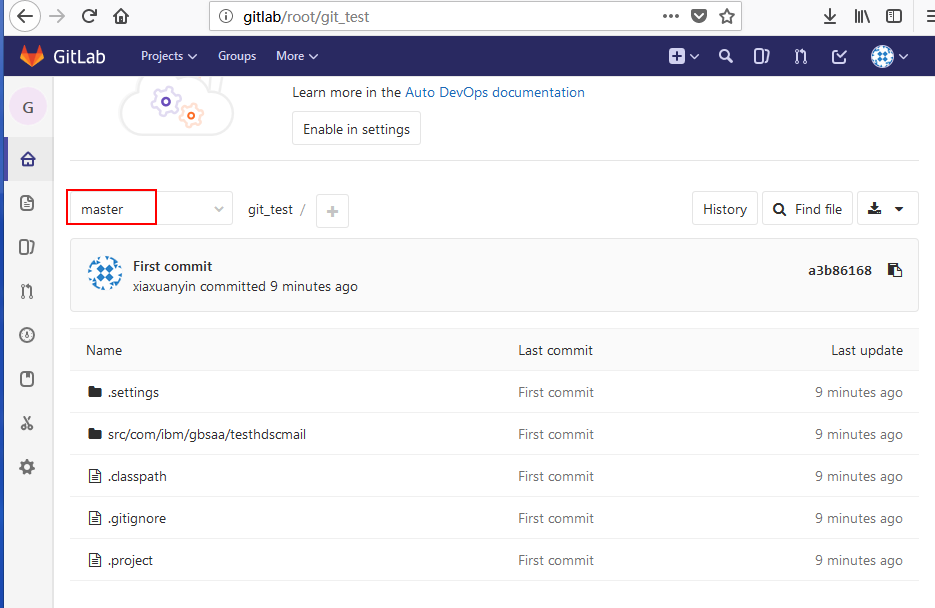
## Git commit 命令

当我们已经完成文件更新到 index 树后，我们可以运行 git commit 命令把更新发布到本地仓库 HEAD 里，如下图将把更新发布到本地主分支。

-m 后接 commit message 参数，本例子是 “First commit”。

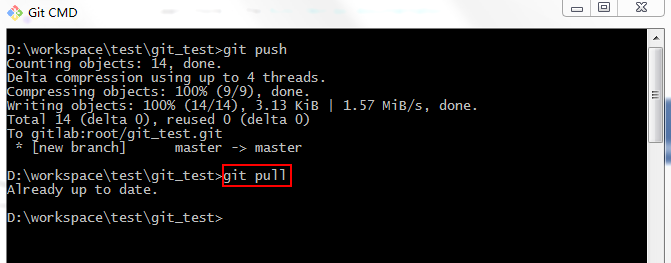
## Git push 命令

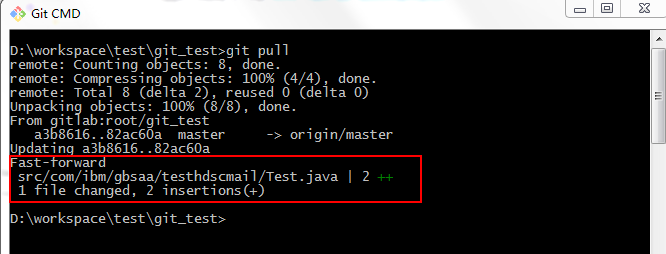
要把本地仓库的更新发布到远端服务器，可以使用 git push 命令。

访问 http://gitlab.hdsc.com , 查看服务器仓库情况，如下图

## Git pull 命令

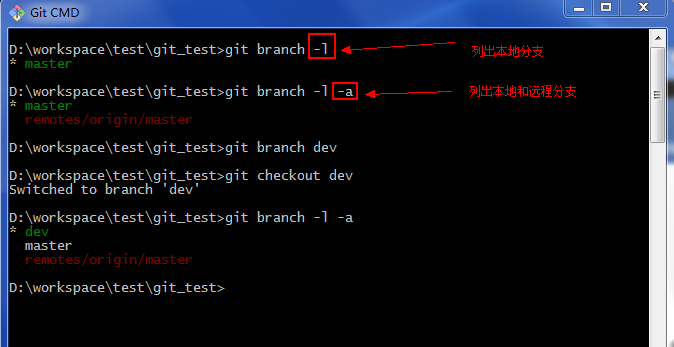
通常一个项目可能会有多个人参与开发，因此我们本地的代码有可能不是最新的，一个好的经验是在每次push 之前都先从服务器 pull 最新的代码到本地，如果有必要，还需要手工 merge，再提交代码到服务器。

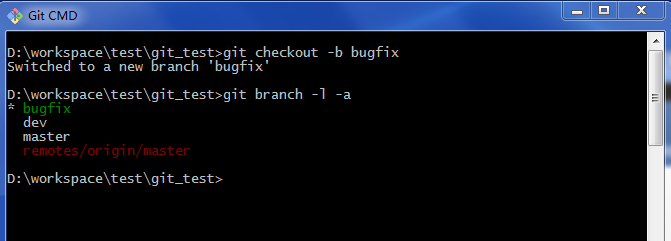
如果本地代码已经是最新的， git pull 结果如图

否则会从服务器下载最新的代码到本地库，如下图

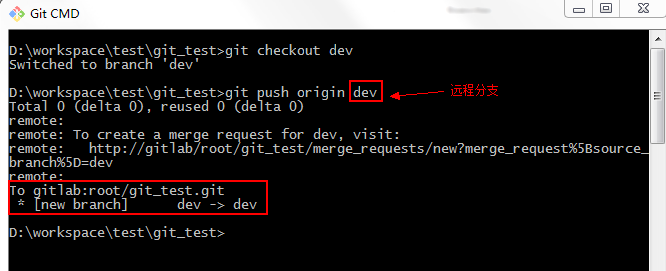
## Git branch 命令

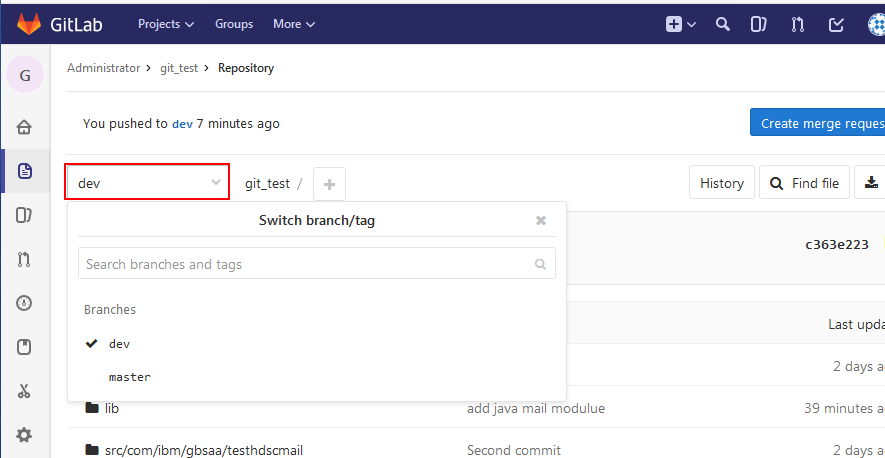
分支是用来将特性开发绝缘开来的。主分支（master）是缺省的，在服务器端除了主分支外，还有 dev 分支和 bugfix 分支。每个项目的主分支只有项目负责人才可以操作，比如合并代码。Dev 分支是开发人员push 和 pull 代码的分支，bugfix 分支只要用于修正测试过程中发现的问题。

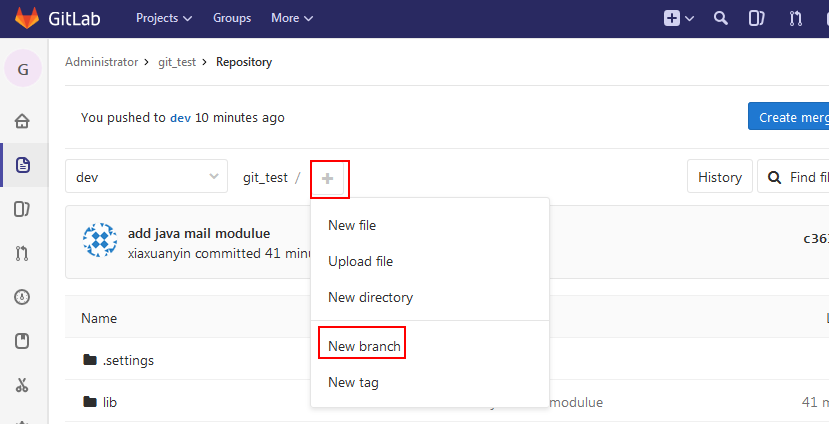
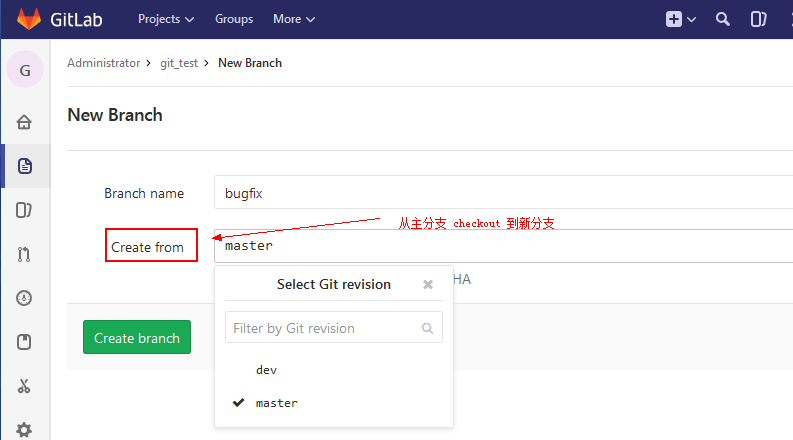
要创建一个本地分支，可以使用如下命令

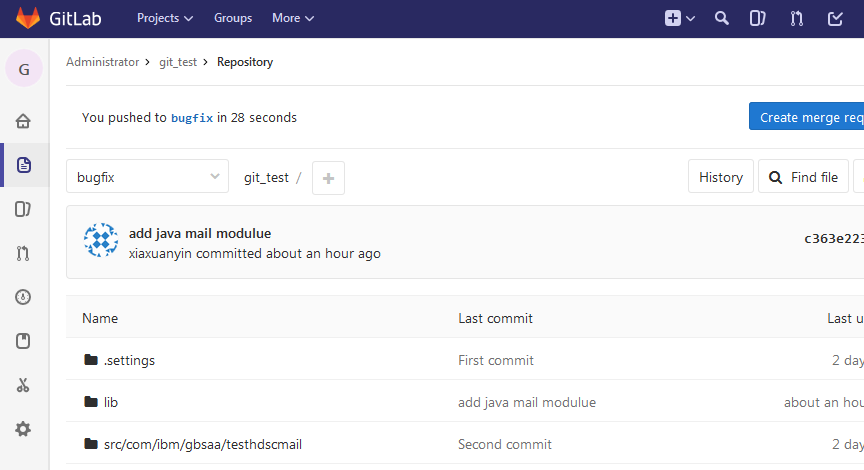
也可以直接用如下命令创建一个新的本地分支并切换到该分支

要删除一个本地分支，可以先切换到 master 分支，然后运行命令 git branch -D <分支名>。

要把本地分支推送到远程服务器，可以用带参数的 push 命令，如下

访问 , 我们可以看到在服务器现在已经有了两个分支，一个主分支，一个 dev 分支，我们可以在下图的红色下拉框里切换不同的分支

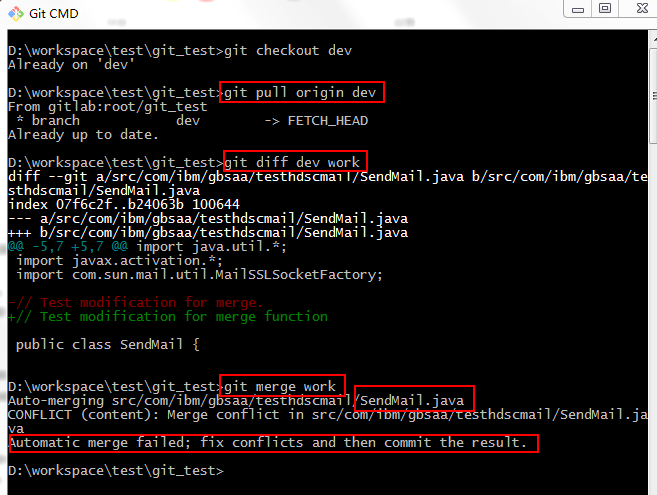
也可以如下图所示，点击红色框的”+”，点击“New branch”子菜单直接创建服务器上的分支 

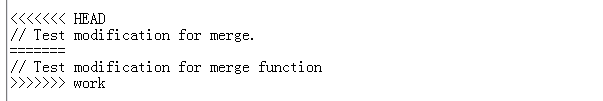
点击 “Create branch”, 创建新的服务器分支 bugfix，如下图

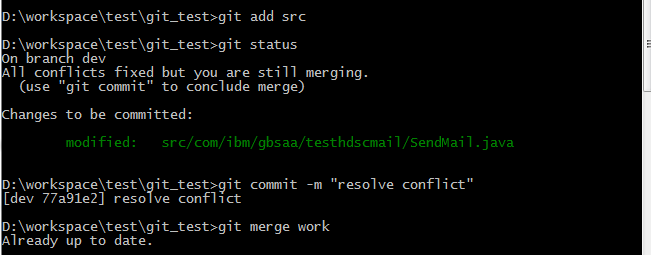
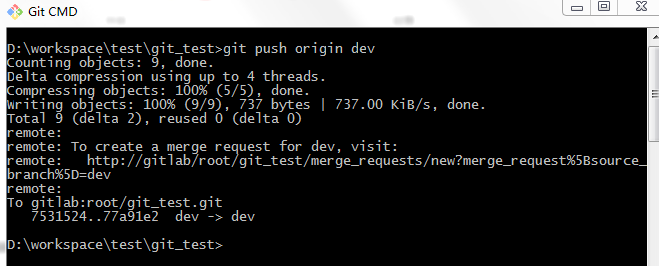
## Git merge 命令

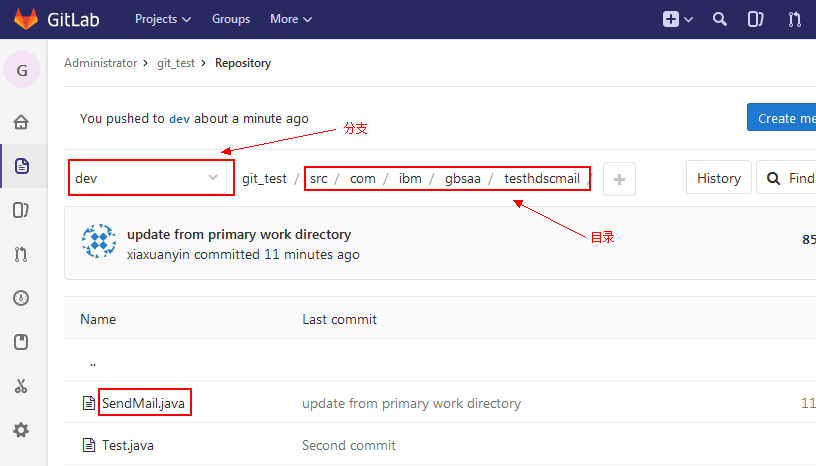
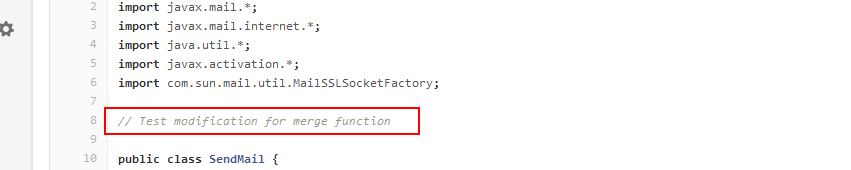
由于服务器端 dev 是我们平时 push / pull 代码的主要分支，为了避免和其他人的代码冲突，我们需要遵循如下的规范。

* 在每次提交代码到服务器 dev 分支时，都要从服务器 pull 最新代码到本地 dev 分支
* 平时工作时，在本地创建一个新的分支， 比如 work
* 在 work 分支完成后，可以运行 git diff 命令 比较 work 分支和 dev 分支的差别
* 切换到本地 dev 分支
* 运行 git merge 命令合并 work 分支到 dev 分支，有冲突的时候需要手动合并代码
* Push 代码到服务器 dev 分支

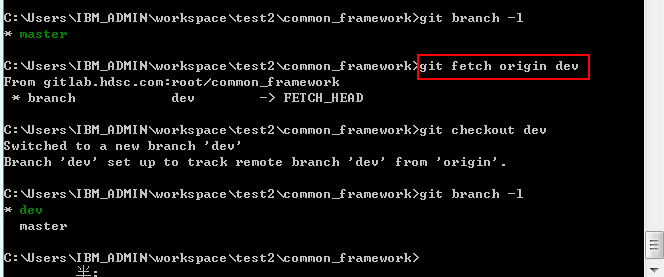
下图尝试去合并 work 和 dev 分支，不幸的是自动合并失败

编辑 SendMail.java, 定位到冲突的地方，如下图。

手动合并代码，保存后，运行 git add 和 git commit 命令，如下图。 

访问 , 检查服务器端代码是否正确，如下图。 

## Git fetch 命令

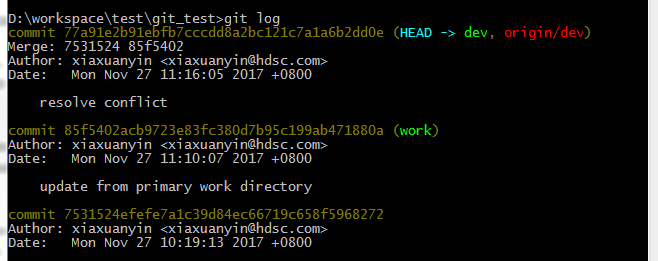
当我们只想从服务器 checkout，又不想让 Git 自动 merge 的时候，我们可以使用 Git fetch 命令，而 Git pull 相当于 git fetch 之后再 git merge。

开发组长在要合并代码到主分支的时候，可以先 clone 一个本地仓库，然后 fetch dev 开发分支， checkout dev, 再 checkout master, 运行 merge 命令。

## Git tag 命令

我们可以使用 git tag 命令在适当的时候给代码打上标签，尤其在我们要发布新版本的时候。步骤如下：

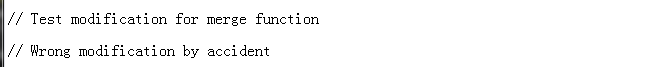
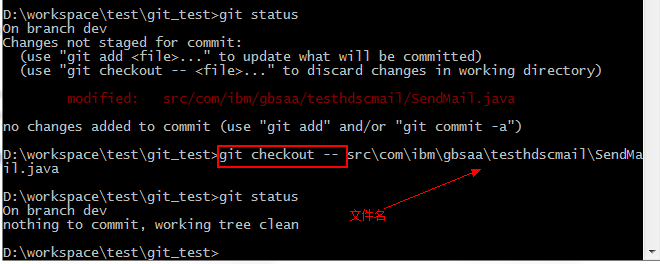
* 运行 git log 从 HEAD 树里查找到要标记的 ID
* 运行 git tag 命令。



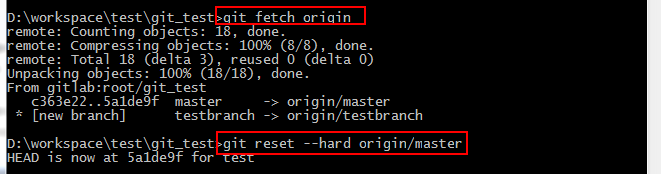
找到 id 77a91e2b91ebfb7cccdd8a2bc121c7a1a6b2dd0e (通常前10位就可以，只要是唯一的)，运行git tag 命令，如下图。

C:\Users\XIAXUA~1\AppData\Local\Temp\1511754344(1).png

## Git 替换本地改动

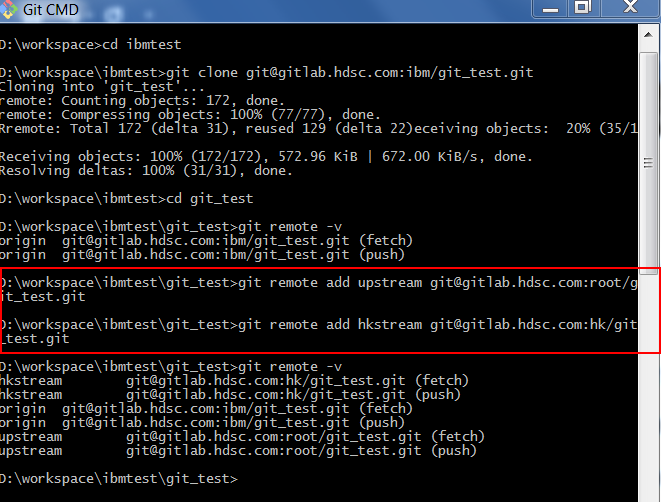
如果我们要放弃某个文件的本地改动，可以使用如下命令

查看该文件，发现错误的改动已经被恢复。C:\Users\XIAXUA~1\AppData\Local\Temp\1511755280(1).png

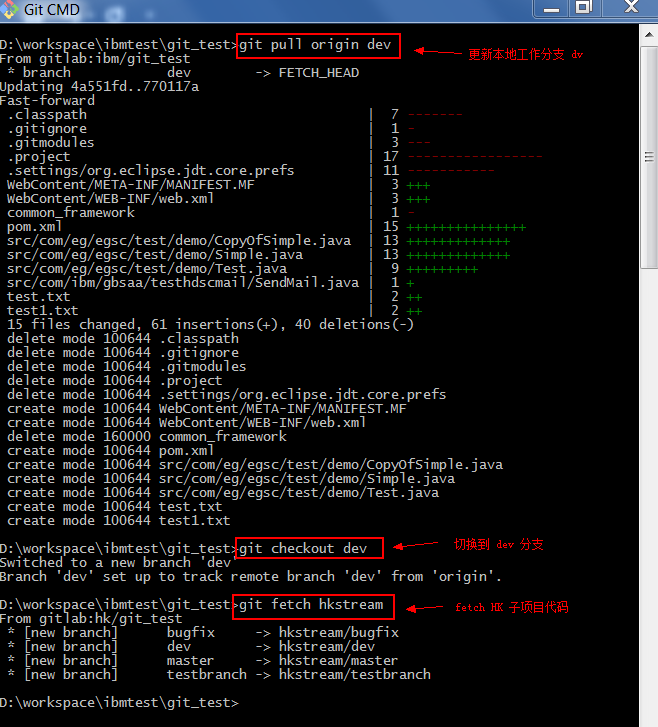
如果要放弃所有本地的改动，可以使用如下命令

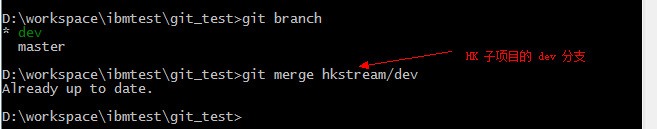
## Git 父项目和 fork 项目间的合并

有时同一个项目可能有多厂商同时参与开发工作，为了避免彼此间的干扰，将从父项目 fork 不同的子项目，下面以 git\_test 举例说明。

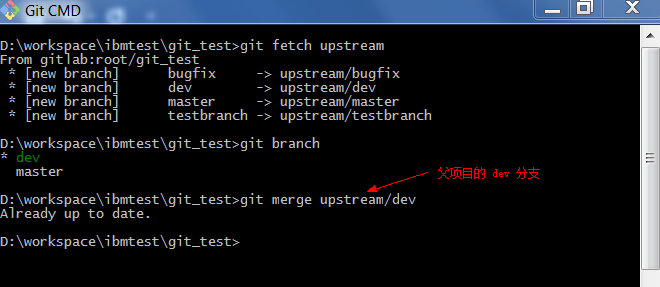
一下假设我的子项目是 IBM 子项目。先使用如下命令创建 IBM 子项目的本地仓库，并指定父项目及 HK 子项目的远程库。

### 合并其他子项目

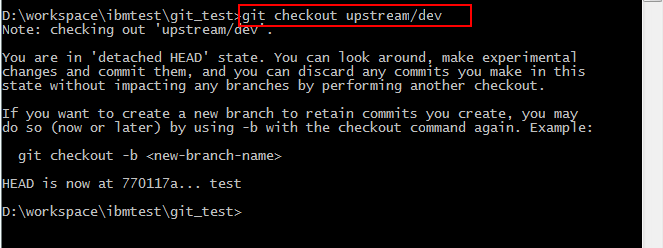
通常会设定子项目负责人可以从其他子项目 pull 代码，但是不可以 push 到其他子项目。如果我们要把HK子项目的最新代码合并到IBM 子项目，可以使用如下命令

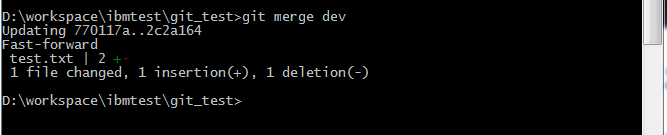
准备合并

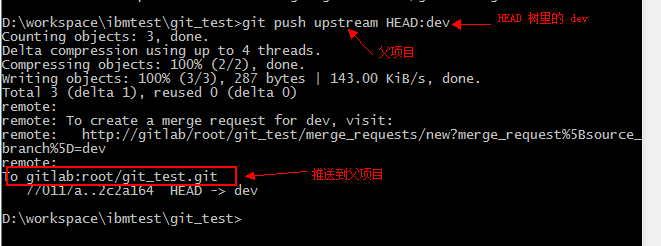
### 合并父项目代码到子项目

和合并子项目类似，命令如下

### 合并子项目代码到父项目

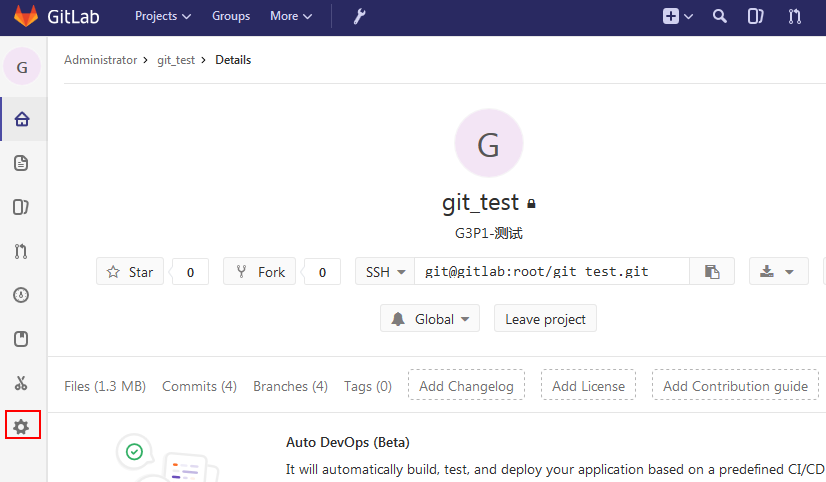
首先，切换到父项目的工作分支 dev，

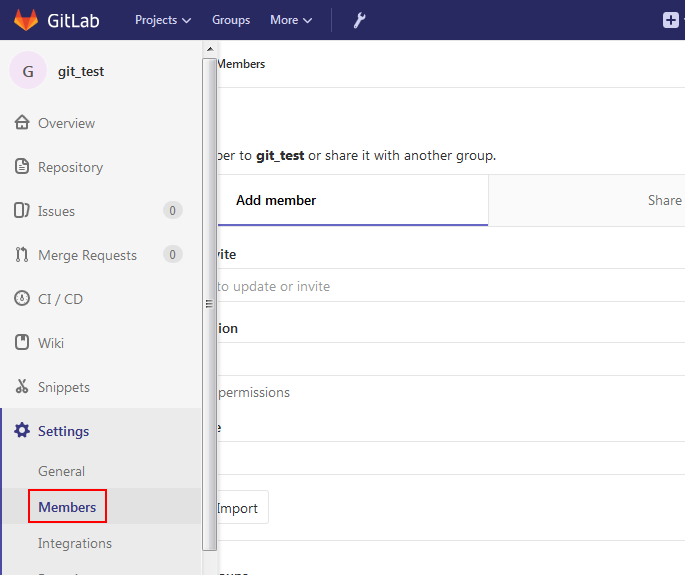
合并IBM 子项目的 dev 分支

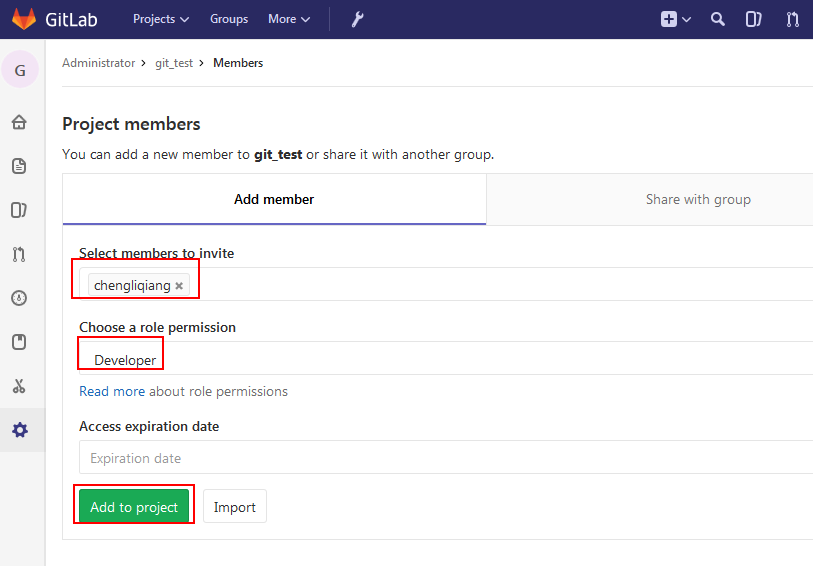
Push 到服务器父项目 dev 分支

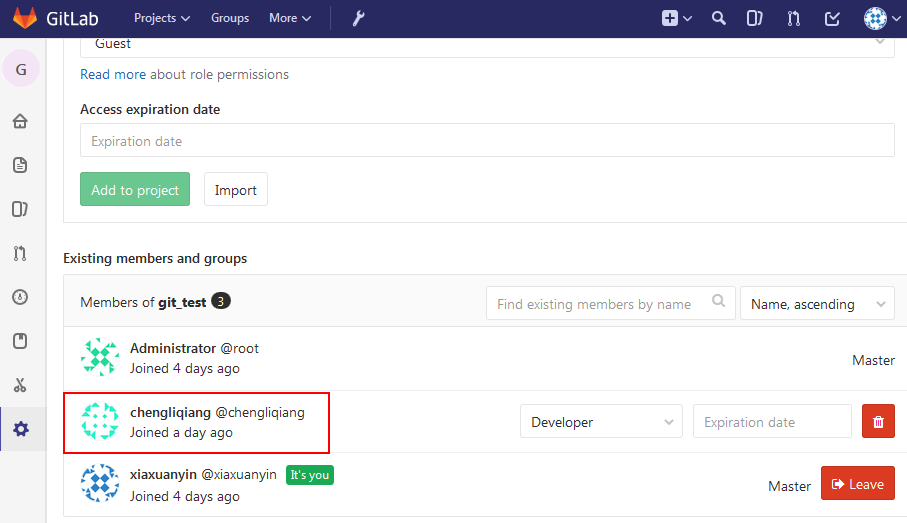
# Git Web 界面

## 添加用户

登录 [，点击项目，再点击左侧导航](http://gitlab.hdsc.com，点击项目，再点击左侧导航) settings，如下图红色齿轮

点击 Members 子菜单

在“Select members to invite”输入框里输入要加入的用户名，在“Choose a role permission”输入框里选择 Developer, 点击“Add to project”按钮，如下图

点击下拉滚动条，可以看到用户已经被加入到项目。

# Git GUI