Git使用说明

1. 基本配置

在github中新建用户

在github中新建组织

添加其他用户

1. 代码操作
   1. 下拉代码

git clone 链接

此处链接可以是https地址路径也可以是ssh地址路径

下拉结束后，若需要后续开发提交代码，则需要在项目中进行配置，

执行：git init，

如果要取消初始化配置，则执行：rm -rf .git/

* 1. 代码提交
     1. git add .
     2. git commit -m “提交说明”
     3. git remote add origin 链接

在提交时如果报错：fatal: remote origin already exists.

先输入：git remote rm origin

* + 1. git push origin master

如果在提交时提示：error: failed to push some refs to 'https://github.com/gis-coder/gis-poi.git'

可以通过如下命令进行代码合并 注：pull=fetch+merge

git pull --rebase origin master

* 1. 更新代码

git pull

项目代码连接：<https://github.com/gis-coder/gis-poi.git>

* 1. 强制更新本地代码（与远程仓库保持一致）

git强制覆盖：

git fetch –all

git reset --hard origin/master

git pull

git强制覆盖本地命令（单条执行）：

git fetch --all && git reset --hard origin/master && git pull

第一个是：拉取所有更新，不同步；

第二个是：本地代码同步线上最新版本(会覆盖本地所有与远程仓库上同名的文件)；

第三个是：再更新一次（其实也可以不用，第二步命令做过了其实）

* 1. 修改github日志
     1. 情况一：最后一次提交且未push

执行以下命令：

git commit --amend

git会打开$EDITOR编辑器，它会加载这次提交的日志，这样我们就可以在上面编辑，编辑后保存即完成此次的修改。

* + 1. 情况二：最后一次提交且已push到服务器

执行以下命令：

git commit --amend

git push origin master --force

和情况一的做法一样。使用push推送到远程服务器是需要加上--force，让服务器更新历史记录。

需要注意的是：把修改后的日志强制push到Git服务器，如果别人本地的副本有修改，很有可能会导致他们同步不了，所以最好和他们核对下。

* + 1. 情况三：旧的提交且未推送

假设commit是倒数第3次提交，这个可以使用git log查看，

$ git log

commit b1b451d218cc23b6c769f373164f2b89cf54d0aa

Author: clcaza <clcaza@sina.cn>

Date:  Sat Mar 10 19:09:08 2018 +0800

  添加内容d

commit 04f0d1809d5d31cc6e930efcba47a5f3f7e93319

Author: clcaza <clcaza@sina.cn>

Date:  Sat Mar 10 19:08:24 2018 +0800

  添加内容c

commit 94fc8feb916442d56b558d5c370f18f057298921

Author: clcaza <clcaza@sina.cn>

Date:  Sat Mar 10 19:07:08 2018 +0800

  添加内容a

commit fd517efa9faf6a5ec71d0eac38fbcfa0cd689f40

Author: clcaza <clcaza@sina.cn>

Date:  Sat Mar 10 19:06:21 2018 +0800

  init

执行rebase

git rebase -i HEAD~3

它会打开一个编辑器，它会把最后前3次的提交显示出来，类似于：

pick 94fc8fe 添加内容a

pick 04f0d18 添加内容c

pick b1b451d 添加内容d

你会看到，它是按提交的顺序显示的，与git log显示的顺序相反。定位到你要编辑日志的那一行，把pick修改为edit，然后保存。

接着就是修改日志内容了

git commit --amend

完成编辑日志后，记得执行：

git rebase --continue

Rebase目的是打开提交的历史记录，让您选择要修改的内容。 Git会让你在一个新的分支修改内容。 git rebase --continue则是让你重新回到之前的分支。

* + 1. 情况四：旧的提交且已push到服务器

前面编辑日志的操作是和情况三是一样的：

git rebase -i HEAD~X

git commit --amend

git rebase --continue

X表示倒数第几次提交。

完成编辑日志后，执行push：

git push origin master --force

* 1. Vim编辑器使用

i：进入编辑状态，即输入模式

esc：退出编辑状态，即进入命令模式

x：删除当前光标所处的字符

: ：切换到底线命令模式，以在最底一行输入明林

w：保存文件

q：退出程序