Git工具进阶

1. 高级配置git

本地端的.gitignore文件操作。

.gitignore文件的作用：在使用某些本地工具进行网页的开发或者是其他代码的开发会自动在仓库的代码空间里自动生成某些隐藏文件，而我们在使用git add命令将本地文件进行提交的时候不想提交相关的自动生成文件就可以使用.gitignore文件。如果不使用.gitignore文件进行忽略的话，使用git status会出现.开头的虚拟文件没有提交（红色）的提醒。为了避免每次修改文件后不必要且不太重要的提交，我们使用.gitignore。

.gitignore的使用方法：

适用场合：

1. 忽略操作系统自动生成的文件，如缩略图的产生
2. 忽略编译器产生的中间文件，如c编译器在编译过程中生成的.obj文件.exe文件
3. 忽略某些隐私文件，比如自己的密码等。
4. /tmp临时目录
5. /log目录

使用方法：在仓库目录下面新建一个.gitignore文件，在里面添加要忽略掉的文件或一类文件（一行表示一个文件或一类文件）

注意：.gitignore文件不可以被忽略。

要想将添加进.gitignore中的文件添加到暂存区可以使用：git add –f 文件名

实例：

第一步：在当前仓库下面新建一个.project文件，在里面随便添加几句话就可以，保存退出。

第二步：使用git status查看当前仓库的状态，会发现.project文件为红色，表示在本地端

第三步：在当前仓库下面使用vi .gitignore文件 ，并在在文件中添加一行.project，保存退出

第四步：使用git status查看仓库状态。此时只有.gitignore文件为红色，因为其不能忽略自己。但是.project已经完成被忽略了。

第五步：将.gitignore提交到暂存区，再提交到仓库就可以了。以后要添加忽略的文件或某一类文件直接修改.gitignore文件就行了

远程端github网站端的.gitignore的配置。

在github上面新建一个仓库，在新建的时候添加.gitignore文件。在下拉列表里面有很多工程对应的.gitignore文件，自己根据搭建的工程类型选择

就行。如果不满足这些类型里面的忽略文件内容，可以自己在线修改保存提交就可以了。知识点：.gitignore文件是在github上托管的一个仓库。

一些小细节：

在windows下面的git bash命令行下使用git add 文件是会出现一行警告信息，这个是什么原因，怎么去掉呢？

原因：

CR: carriage return ,回到行首，回车，\r

LF： line feed 跳到下一行 ，/n

Linux下面的换行符：/n 跳到下一行

Windows下面的换行符：/r/n 回到行首在换行。

Mac 下的换行符：/r

上面出现警告的原因就是git bash操作为linux的操作，而我们编写的文件是在windows下面进行编辑的，所以会存在警告。

解决方法：

git config –global core.autocrlf ture 使能在进入git bash时自动将crlf换成lf

git config –global core.safecrlf flase 允许提交包含混合换行符的文件，这里设置为失能，文件总不能在windows下又在linux下进行编辑对吧。

别名与缓存凭证：

别名其实和linux bash的别名设置是一样的

使用方法一：git config –global alias.ci commit 这样就将commit命令别名设置为ci了

使用方法二：在主文件夹下，cd ~ ，vi .gitconfig文件，在[alias] 栏目下进行设置，遵循shell变量语法就行。

缓存凭证设置：作用，不用每次在向远程推送本地仓库时输入用户名和密码

操作方法一：如果本地和远程的传输协议继续使用https协议的话：git config –global credential.helper wincred 这样就可以了

操作方法二：可以使用其他的文本传输协议，如使用ssh协议

1. git的协议

ssh传输协议：

在远程的github网站端，我们克隆一个仓库到本地可以使用ssh协议也可以使用https协议。

https协议与ssh协议的对比：

1. https协议在本地到远程的推送时需要输入用户名和密码如果没有进行git config –global credential.helper wincerd
2. https协议在进行远程推送的时候推送大小为原本大小，但是ssh协议使用的是压缩后的文件大小，所以ssh的速度更快

三、ssh比较安全，但是某些企业内部使用的平台可能会关掉ssh协议使用的公开端口（也有可能防火墙不允许），即不会使用ssh服务，此时就需要

使用https协议了

怎么使用ssh协议作为git的传输协议呢？

Ssh协议下的基本命令：

git clone ssh://git@github.com/user.name/user.repos

git clone [git@github.com:user.name/user.repos](mailto:git@github.com:user.name/user.repos) //注意，这里在user.name前使用的是冒号

上面两个命令均是将远程的仓库克隆到本地

Git remote add [shortname] [url] shortname一般为origin [url的格式git@github.com:uesr.name/user.repos](mailto:url的格式git@github.com:uesr.name/user.repos) 将本地仓库与远程仓库建立关联。

查看当前仓库使用的传输协议，注意传输协议是针对具体的一个用户的（即github网站上的一个用户），所以设置好所有的仓库都可以使用率

Git remote –v ，若显示出来的右https则使用的传输协议为htpps协议，[如果右git@github.com就为ssh](mailto:如果右git@github.com就为ssh)协议

使用ssh协议的方法：（实例）

第一步：在git bash命令行中输入 ssh–keygen –t rsa –C “your email”.这样就生成了一对用于ssh的公钥与私钥，这些秘钥存储在主目录下的隐藏文件.夹ssh下。

第二步：进入到.ssh文件夹下使用cat ssh\_rsa.pub 得到公钥，将公钥复制拷贝到github的个人设置的SSH and GPG Keys中。此时钥匙为黑色的。

（注意：可以将这个公钥与私钥保存下来，这样将公私钥拷贝到另外一台电脑上就可以登录ssh）

第三步：此时目录为.ssh，需要切换到存储仓库的地址，cd /g/git\_re,使用git clone [git@github.com：larioy/demo](mailto:git@github.com：larioy/demo) 这样就使用ssh协议将远程demo仓库克隆下来了

第四步：使用git remote –v查看选用的传输协议

1. 深入讲解常用的git命令  
   git /git help –a 可以查看git的所有命令

Git blame 文件名 可以查看同一文件的在什么时候被谁提交过几次，这个可以用来问责某一部分人写的代码。可以使用管道与grep命令结合使用

Git status –sb可以省去其他的信息，直接列出文件名，这样比较简洁。

Git add的深入讲解：

在git bash命令行下使用Bash命令对文件、文件夹的操作，结果是在本地进行操作。即使用git status查看状态的话，会为红色

如果使用git bash命令对文件与文件夹操作，那么操作的层次是在暂存区中，即自动保存到暂存区中（相当于自己增加了add命令），用git status查看时对应的为绿色。

可以在.gitconfig中设置别名，ad add . 一次性把所有修改过的文件进行提交。

Git add对一个文件进行多次提交（即每一次提交一部分）

Git add –p 文件名 在显示出来的提示语中输入s进行分割，接着操作就行了。 需要配合使用git diff命令

Git commit的深入理解

基本的操作语句

提交的时候要注意提交的颗粒度、独立性、相关性

---以一个小功能小单位或一个bug fix为单位

---一个完整的test一般要在一个提交中

---无相关的修改不要在一个提交中

---语法错误或者半成品不要进行提交，即提交的必须为完整的，无错误的。

Commit提交的语法的基本格式。

Git 的信息查看

Git status

Git log –n 显示想要查看的条数

Git log | grep “word” 从显示信息里抓取想要的信息。

Git diff详解

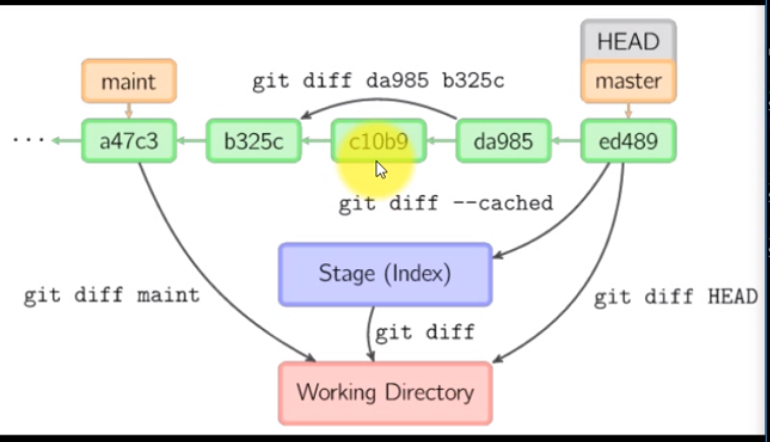
Git diff显示出本地文件与暂存区文件的而不同

Git diff –cache 显示暂存区与最新版本之间的区别

Git diff HEAD 显示最新版本与本地的而区别

Git diff –cache HEAD^ 显示暂存区与倒数第二个的区别 HEAD~3

Git diff HEAD HEAD~3 显示最新版本与最新倒数第4个版本的区别。



1. git的tag工具

git的tag工具就是给每次修改的版本的哈希值设置一个别名，这个命令需要配合git log 命令使用来查看每次提交版本的哈希值。

设置方法：git tag 别名名称 某一个提交的版本的哈希值。

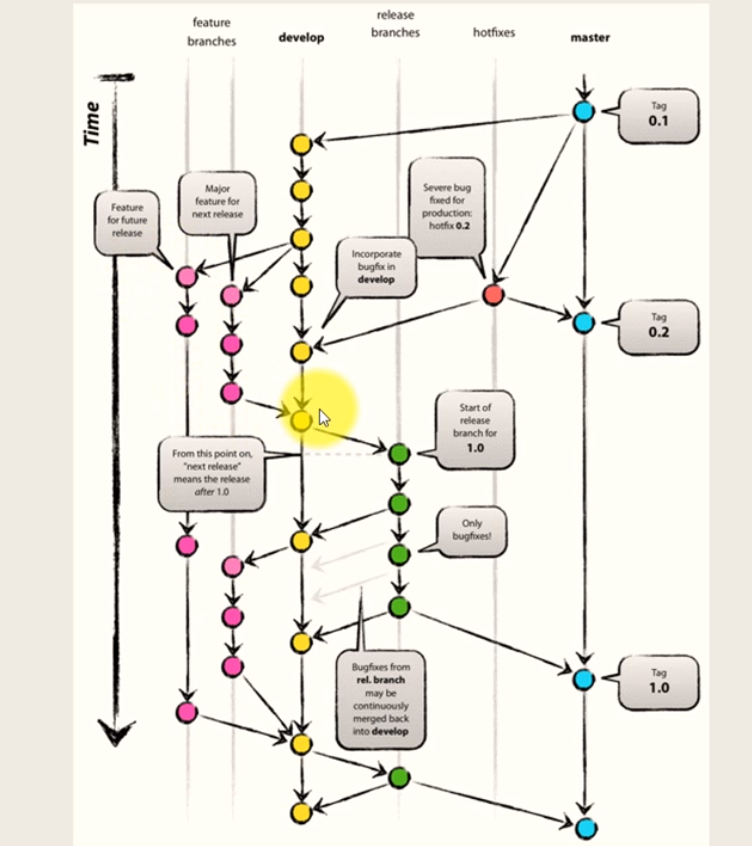
删除tag的方法：git tag -d 别名名称

将某一个把别名推送到远程的方法： git push origin 别名名称。 在远程github网站对应仓库下的release菜单下就会看到更改。

将所有的别名都推送到远程中的方法：git push origin tagname1 tagname2…..等等。

查看当前仓库有哪些别名：git log 或者 git tag（这个显示比较简单）

1. git的分支操作

分支的理由：分支操作可以实现并行开发

相关的命令：

创建一个新分支： Git branch 分支名

切换到另外一个分支：git checkout 分支名。 如果在切换到另外一个分支时，当前分支的内容修改了但是没有提交，需要使用git stash暂存

一步做到创建与切换：git checkout –b 分支名

修改分支的名字：git branch –m/-M old\_name new\_name

删除分支：git branch –d 分支名

查看分支信息： git branch / git branch -v

查看远程分支：git branch –r

查看本地已经合并的分支：git branch –merged

查看远程已经合并的分支：git branch –r --merged

取出远程分支： git checkout –t 分支名

合并分支：git merge 分支名（注意需要切换到主分支以后在进行该操作）