Git的简单入门

虽然久闻Git的名号，但工作用着ClearCase也还是感觉良好。在GitHub上也用过，它提供的GUI工具很直观方便，便一直没有关注命令行的使用。这几天试用一下OSC的Git私有库，发现还是Git的命令行好用些。一些简单的摸索试用，现在终于创建项目成功了。记录一下，希望能够方便一些人。

首先我们需要在[Git@OSC](mailto:Git@OSC)上创建一个项目，假定项目地址为<https://git.oschina.net/sulliy/i.git>。然后按照Google的Git。

在自己机器上，选择一个项目的文件夹，可以是已经有实现的。如果Git已经有了右键关联，可以直接右击文件夹选择Git Bash。没有关联可以先启动Git Bash，然后到项目的目录下去。有一点，不同于Cygwin，盘符的选择在Git Bash是通过”cd e:”实现的。

在Bash中输入下面命令，初始化Git：

$ git init

Initialized empty Git repository in c:/Users/C/Workspaces/MyEclipse 7.0/i/.git/

然后我们Clone下在[Git@OSC](mailto:Git@OSC)的项目：

$ git clone <https://git.oschina.net/sulliy/i.git>

Cloning into 'i'...

Username for 'https://git.oschina.net': [i@sina.com](mailto:i@sina.com)

Password for 'https://i@sina.com@git.oschina.net':

remote: Counting objects: 3, done.

remote: Total 3 (delta 0), reused 0 (delta 0)

Unpacking objects: 100% (3/3), done.

然后我们就可以在自己的文件夹下创建自己的内容了。然后将文件夹下面的所有文件加到Git中（也可以指定某个文件）：

$ git add .

warning: LF will be replaced by CRLF in .mymetadata.

The file will have its original line endings in your working directory.

提交代码最好带上自己的编辑签名，所以现在设置一下Git：

$ git config --global user.name "I"

$ git config --global user.email [i@sina.com](mailto:i@sina.com)

为了能提交代码，我们还需要SSH的私钥和公钥。我们退后一个目录，然后生成SSH的私钥和公钥：

$ cd ..

$ ssh-keygen -C ['i@sina.com'](mailto:'i@sina.com') -t rsa

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/C/.ssh/id\_rsa): key

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in key.

Your public key has been saved in key.pub.

The key fingerprint is:

41:11:0b:6a:77:2d:be:3c:8d:09:2d:69:58:64:44:2f [i@sina.com](mailto:i@sina.com)

现在我们可以在目录下看到两个文件，一个是key，一个是key.pub。将公钥文件打开，并把所有的内容设置到[Git@OSC](mailto:Git@OSC)的公钥里。

到现在我们可以尝试提交一下代码了：

$ cd i

$ git commit -m "test commit"

[master (root-commit) 299f7af] test commit

warning: LF will be replaced by CRLF in .mymetadata.

The file will have its original line endings in your working directory.

11 files changed, 121 insertions(+)

create mode 100644 .classpath

create mode 100644 .mymetadata

create mode 100644 .project

create mode 100644 .settings/.jsdtscope

create mode 100644 .settings/org.eclipse.jdt.core.prefs

create mode 100644 .settings/org.eclipse.wst.jsdt.ui.superType.container

create mode 100644 .settings/org.eclipse.wst.jsdt.ui.superType.name

create mode 100644 WebRoot/META-INF/MANIFEST.MF

create mode 100644 WebRoot/WEB-INF/web.xml

create mode 100644 WebRoot/index.jsp

create mode 160000 i

这个提交并没有真正提交到[Git@OSC](mailto:Git@OSC)的。Git的仓库分为：远程仓库(blasd(remote) repository）和本地仓库（local repository），本地仓库又分为：工作区（workspace），中间状态（staged），本地仓库（local repository）。上面那个提交只是提交到了本地仓库。

在clone完之后，git会自动为此远程仓库命名为origin，origin只是为远程仓库起的一个别名；并且git会建立一个指向远程仓库origin的master 分支的标识，git使用(远程仓库名)/(分支名) 这样的形式表示远程分支，所以origin/master指向的是一个我们clone到本地的远程分支，被代表为远程创仓库和分支。同时，git会建立一个本地master分支，它指向的是从远程仓库的分支下载到本地的副本。可以通过下面的命令查看分支：

$ git branch -a

\* master

remotes/origin/master

我们现在尝试把代码提交到远程仓库。

这里介绍一个命令：

$ git remote –v

它显示远程仓库的信息。这样可以检查自己的Git是否有远程仓库信息，不然对远程仓库的Pull和Push都肯定会失败。

执行上面命令，发现没有远程仓库的信息。先添加远程仓库：

$ git remote add origin <https://git.oschina.net/sulliy/i.git>

再次执行，可以看到远程仓库的信息了：

$ git remote -v

origin  https://git.oschina.net/sulliy/i.git (fetch)

origin  <https://git.oschina.net/sulliy/i.git> (push)

现在试试提交代码：

$ git push origin master

Username for 'https://git.oschina.net': [i@sina.com](mailto:i@sina.com)

Password for 'https://i@sina.com@git.oschina.net':

To <https://git.oschina.net/sulliy/i.git>

! [rejected]        master -> master (non-fast-forward)

error: failed to push some refs to 'https://git.oschina.net/sulliy/i.git'

hint: Updates were rejected because the tip of your current branch is behind

hint: its remote counterpart. Merge the remote changes (e.g. 'git pull')

hint: before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

唉，发现远程的仓库要新一些，我们需要先做个Merge。

$ git pull i

From i

\* branch            HEAD       -> FETCH\_HEAD

Merge made by the 'recursive' strategy.

.gitignore | 6 ++++++

1 file changed, 6 insertions(+)

create mode 100644 .gitignore

现在我们可以将自己的本地仓库代码推给远程仓库了。

$ git push origin master

Username for 'https://git.oschina.net': [i@sina.com](mailto:i@sina.com)

Password for 'https://i@sina.com@git.oschina.net':

Counting objects: 19, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (13/13), done.

Writing objects: 100% (18/18), 2.68 KiB, done.

Total 18 (delta 2), reused 0 (delta 0)

To <https://git.oschina.net/sulliy/i.git>

   28d7ba4..2c63d99  master –> master

现在又有另外一个人需要访问代码了，怎么办呢？大概的步骤是一样的，先Clone，再Pull，在Push。