在实战中来学习git如何托管代码，然后如何对代码进行版本控制853769620

OA系统代码

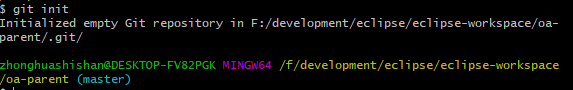
1、创建一个git版本库

有两种方式可以创建一个git版本库，将一个版本的项目目录转变为通过git来进行版本控制；也可以从其他地方克隆一个git版本库。

（1）对一个已有的项目初始化git版本库

我们一般都是先创建一个项目，这个项目处于一个目录下，此时如果要让git来管理这个项目，就可以在这个项目的根目录下面执行以下命令：

git init



执行了git init命令之后，意思就是说，要让git托管这个项目的代码，后续这个项目中的所有代码的变动，都可以作为一个版本提交到git中，git会管理这个项目的每个版本的代码，同时后续我们就可以基于git对这个项目进行各种各样的版本控制的功能

git init命令执行之后，做的第一件事情，此时就会创建一个.git隐藏目录，这里包含了git的所有的文件，就是每个版本的代码都会存储在.git目录中。之前我们不是说，如果要对HelloWorld.java文件进行版本控制的话，实际上是要保留这个文件的多个版本的代码的，如果自己手工做，那么就是保存多个代码文件。

.git目录中，实际上就是存储了你的每个代码文件的每个版本，每个版本就是一个独立的文件。历史版本都被存储在了.git目录中。

但是此时git还没有开始对你的项目进行版本控制。如果要对你的项目代码进行版本控制，需要执行以下命令：

git add --all .

git commit -m 'initial project version'

把上面两个事情给做了之后，实际上就是代表的是，将项目代码当前的版本，提交到git中去，保存到.git目录中去，开始让git管理当前项目的所有代码文件的后续的版本变更

有的时候，我们不是对项目里所有的代码或者是文件都要追踪其版本变更的，尤其是对一些自动生成的一些东西，比如说maven生成的target目录下的东西，跟我们的代码是没关系的

我们是不希望通过git来管理target下面的东西的

我们可以自己手动创建一个.gitignore文件，在里面每行一个，放入不需要被git托管的目录或者文件

纠正一下

我平时工作都是在mac上的

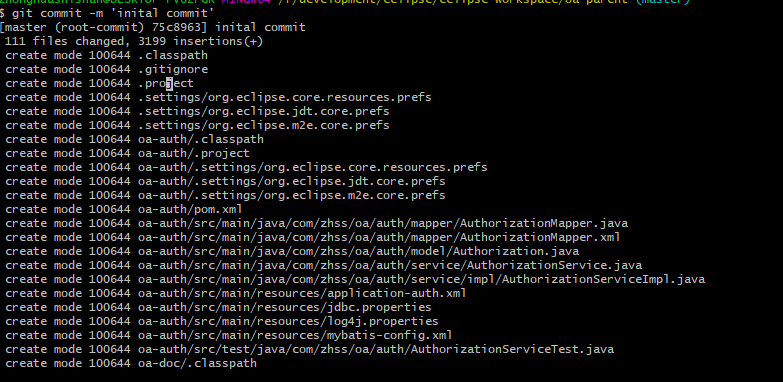
rm -rf .git

git config --global core.autocrlf false

git init

git add --all .

git commit -m 'inital commit'



解释一下，这里就显示出来了，将哪些代码文件提交到了git，一共有多少个文件，有多少行，根据.gitignore排除了所有的target目录下面的东西，不要让git去托管target目录里的内容

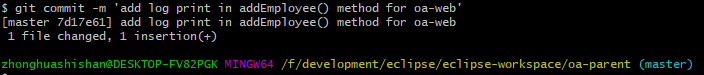
到此为止，我们就对这个项目的所有代码文件都实现了版本控制。

就是我们已经用git托管了oa系统的代码了，oa-parent下所有的代码文件，而且已经提交了第一个版本的代码到git中去了

再来尝试几次，比如在eclipse里面修改我们的代码，修改好了之后，再次提交到git中去，作为第二个版本，再尝试一个第三个版本

就是git commit -m后面跟着的一串内容，就是你这次提交的修改后的代码都干了些什么事情，简短一点

修改oa-web中的代码，加一行日志输出



此时oa-parent这套代码，就形成了第二个版本

再次尝试一下，在oa-auth的server中加入一行日志输出，再次提交

IMG_259

发现说，我们明明就修改了一行代码，2个文件改变了，插入了6行，这个我给大家解释一下，有时候可能是eclipse自动生成的一些文件也变化了

所以我刚才又修改了一下日志打印，然后再次提交

IMG_260

大家会发现说，我们初体验了一把，用git怎么来托管我们的代码，管理代码的多个版本

（1）git init，初步让git准备脱光我们的代码，创建了一个.git目录，其中是git用来存放版本数据的

（2）git add和git commit两个命令，完成了第一个版本的提交，代码提交到了git中，之后所有这些代码文件的变动，都会由git来管理版本

（3）每次我们修改完一块代码，比如开发了一个小功能，或者是修改了一个小bug，就可以用git add和git commit命令，将修改后的最新版本的代码提交到git中

（4）循环往复，通过每次修改代码+提交修改后的代码到git中，git就完成了对我们的每个代码文件的每个版本的数据的存储，.git目录中

（5）接下来，git才能为我们提供各种版本控制的功能