上一讲已经给大家剖析清楚了git仓库结构：工作区，暂存区，仓库853769620

git add和git commit在干嘛

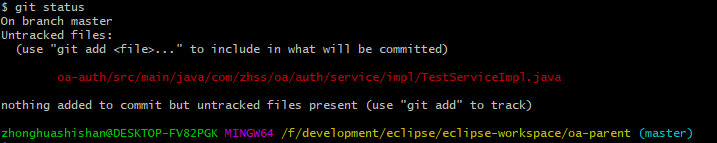
但是结合git中不同的区域，你会发现每一个代码的版本都可能停留在不同的区域中，这就可能处于不同的一个状态下

所以说，这一讲，我们结合上一讲的内容，来学习一下用git status，来查看当前git项目的各种状态

1、动手实验git status查看代码文件的状态（结合git仓库结构）

代码文件分成两种，一种是tracked，一种是untracked。tracked文件就是已经提交到git版本库中的文件，后面可以处于modified或者staged状态；untracked文件，就是从来没有提交到git版本库的代码文件（也从来没有放入暂存区）。

在oa-auth的server包下，可以新建一个TestServiceImpl.java，然后执行git status命令，看一下当前的状态



此时git status会告诉你，有文件没有被提交到git仓库中或者是暂存区中过，就是untracked，就是从来没有被git追踪过它的版本，甚至第一个版本都没有被git追踪。

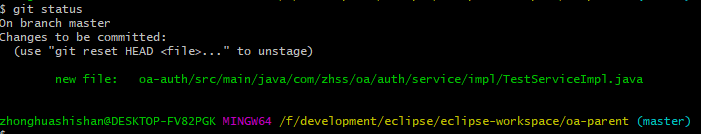
接下来执行命令：

git add oa-auth/src/main/java/com/zhss/oa/auth/service/impl/TestServiceImpl.java

git add --all .，也可以给大家开始来解释一下这些命令的含义，git add .，就是将当前新增或者是修改过的文件，加入暂存区，但是加了--all之后，如果有文件被删除，也会将文件被删除的状态加入暂存区中

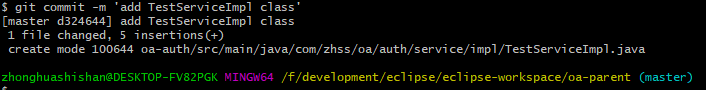
一般工作中，都是用git add --all .

但是也可以直接git add 某个文件，把指定的文件加入到暂存区中



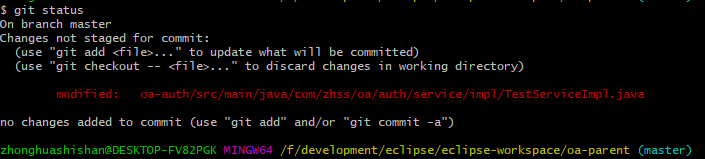
TestServiceImpl.java这个文件，从untracked状态，变成了new状态，就是一个第一次进入git托管的一个新文件，看到的第一个版本

git commit -m 'add TestServiceImpl class'



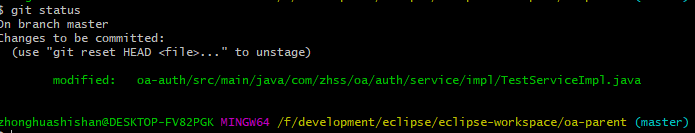
此时，这个文件就进入了一个tracked状态，已经被git开始追踪了

然后对TestServiceImpl.java进行修改，加入一个方法，接着看一下状态

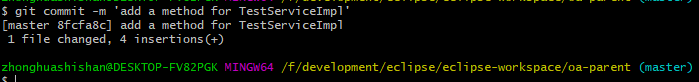


此时变成了modified状态，就是已经被追踪了，但是后来又被修改了，但是是changes not staged for commit，就是被修改了，但是没有被staged，staged就是还没有被加入暂存区

再次执行git add --all .命令



此时的状态是，changed to be committed，modified，此时文件被放入了暂存区，modified状态，同时等待被commit提交到git仓库中去



此时文件变成了committed，已经提交了

总结一下

（1）新文件刚创建：untracked，此时仅仅停留在工作区中

（2）git add 新文件：new file，此时已经被追踪了，放入了暂存区中

（3）git commit 新文件：committed，已经被追踪了，放入了git仓库中

（4）修改那个文件：modified，changes not staged to be committed，没有加入暂存区，被修改的内容仅仅停留在工作区中

（5）git add 修改文件：modified，changes to be committed，修改的文件版本被已经加入暂存区

（6）git commit 修改文件：committed，修改后的新版本提交到了git仓库中

一般自己创建的版本库，刚开始文件都是untracked，然后git add和git commit命令执行之后，就被提交了第一个版本到git版本库，此时就全部都是tracked了。如果是从远程版本库克隆下来的，那么刚开始就是tracked。

接着对tracked的文件修改之后，就是modified；然后对modified文件再执行git add命令之后，就是staged，进入了暂存区；接着执行git commit命令之后，就将暂存区中的文件都提交到了版本库中，此时就是unmodified，and tracked。

2、6种状态映射到3种状态的补充说明

git管理的文件有三种状态：提交状态（committed），修改状态（modified），暂存状态（staged）。

提交状态：我们的文件已经安全的保存在git的本地数据库中了。

修改状态：我们修改了文件，但是还没有提交到git的数据库中去。

暂存状态：将修改后的文件标记为即将通过下一次提交，保存到git数据库中去。

（1）新文件刚创建：untracked，此时仅仅停留在工作区中

（2）git add 新文件：new file，此时已经被追踪了，放入了暂存区中 => staged

（3）git commit 新文件：committed，已经被追踪了，放入了git仓库中 => committed

（4）修改那个文件：modified，changes not staged to be committed，没有加入暂存区，被修改的内容仅仅停留在工作区中 => modified

（5）git add 修改文件：modified，changes to be committed，修改的文件版本被已经加入暂存区 => staged

（6）git commit 修改文件：committed，修改后的新版本提交到了git仓库中 => committed

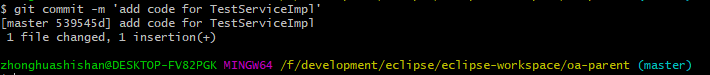
3、补充实验，彻底了解git add和git commit的关系，包括3个区域的关系

（1）首先我们先修改一下TestServiceImpl这个类，加入一行代码

（2）执行一下git add命令

（3）再次修改一下TestServiceImpl这个类，再加入一行代码

（4）直接执行git commit命令



大家必须建立起来一个版本的概念

（5）解释说明

第一次加入一行代码：System.out.println("invoke test method......");

这是一个版本，v3，然后git add之后不，就是带着这一行的v3版本放入了暂存区

接着你又添加了一行代码，System.out.println("prepare to return value......");

这又是一个版本，v4，但是v4，带着System.out.println("prepare to return value......");行代码的版本，没有被放入暂存区

此时直接执行git commit操作，是将暂存区里的v3版本的代码，就只有一行System.out.println("invoke test method......");代码，提交到了git仓库，此时就是认为，就插入了一行而已

git commit操作，只会将已经放入了暂存区的代码提交到仓库

你带着System.out.println("prepare to return value......");行代码的v4版本，还停留在工作区呢，changes not staged to be committed。

你需要再次git add + git commit，才能将v4版本提交到git仓库

总结一下，对于git来说，什么是一个版本？

一次git add算做一个版本，每次执行一次git add操作，都是将工作区中所有修改的文件，都作为一个新的版本，放入暂存区，这个版本等待被提交

为了让大家不要迷糊

最后说明一下，就是实际工作中，一般很少说多次git add，过了很久，搞一次git commit

说真的，一般其实每次git add和git commit都是一起执行的

就是你修改了多少代码，然后你希望作为一个新的版本，去提交，那就一次性git add，然后git commit就可以了，一次git add + git commit，就出来了一个新的版本

必须git add到暂存区里的修改，才会被git commit时提交到仓库里，否则就是停留在工作区中