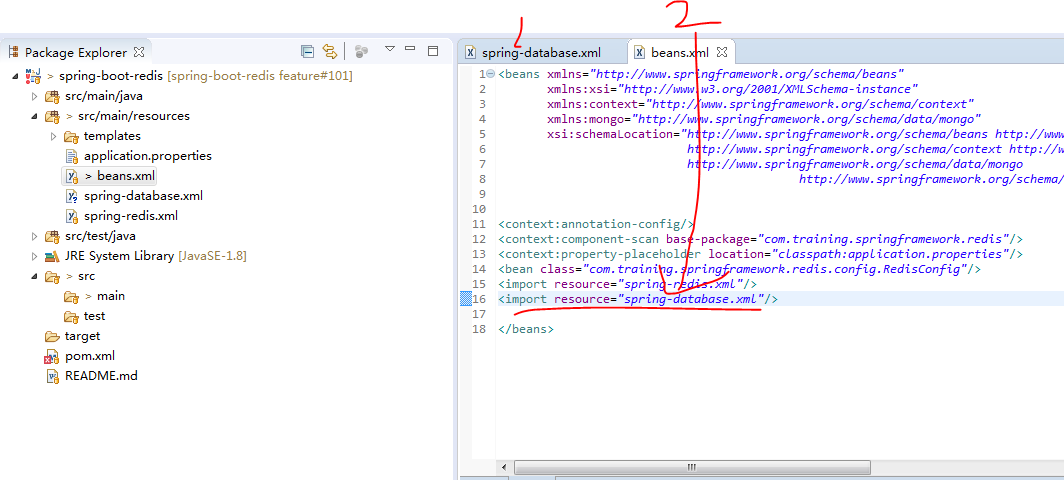
**5,Git stash 案例使用**

git stash 可用来暂存当前正在进行的工作， 比如想pull 最新代码， 又不想加新commit， 或者另外一种情况，为了fix 一个紧急的bug, 先stash, 使返回到自己上一个commit, 改完bug之后再stash pop, 继续原来的工作。

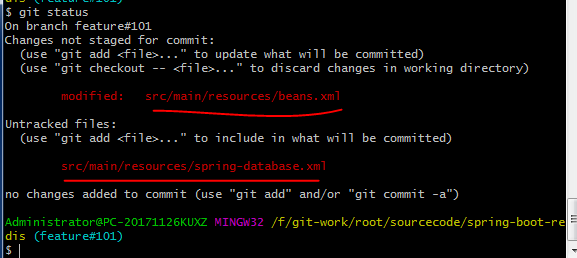
1，案例场景

在feature#101上开发，如： 新增内容如下： 新增一个文件spring-database.xml, 在beans.xml中增加一行。这时，发现原有的分支feature#61上需要修改BUG。



需要在本地切换到feature#61，feature#101本地的未提交的更改需要暂存在git stack中。这时候你的工作区间feature#101和上一次提交的内容是完全一样的，所以你可以放心的修复feature#61分支Bug，等到修完Bug，提交到服务器上后，重新切换到feature#61工作区，再使用git stash apply将以前一半的工作应用回来。

Step1: $git status 查看状态



根据提示： 使用

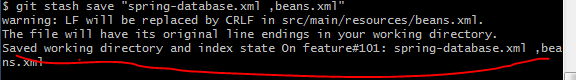
$git checkout – src/main/resources/beans.xml

可以放弃beans.xml文件的修改。

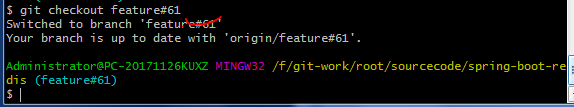
使用：

$git add spring-database.xml 可以将spring-database.xml文件提交到本地。

Step2: $git stash save “spring-database.xml, beans.xml”



Step3, 切换到feature#61 分支工作区

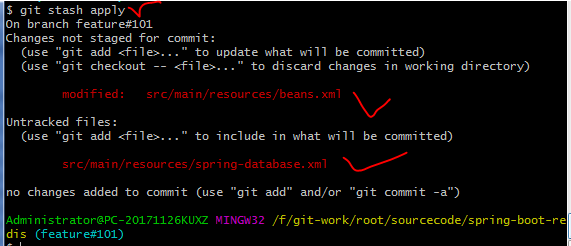


Step4, 修复BUG后重回到 feature#101

$git checkout feature#101

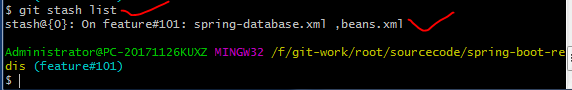
Step5, 将缓存堆栈中的stash多次应用到工作目录中，但并不删除stash拷贝

$git stash apply

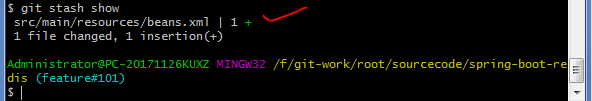


Step5,查看现有stash

$git stash list

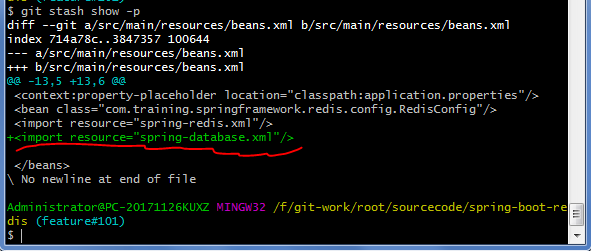


$git stash show



查看全部的diff

$git stash show –p

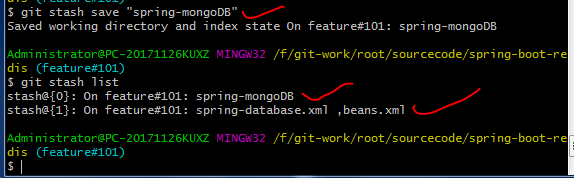


2, git stash pop 和 git stash list

git stash pop : 删除最后一次缓存的stash

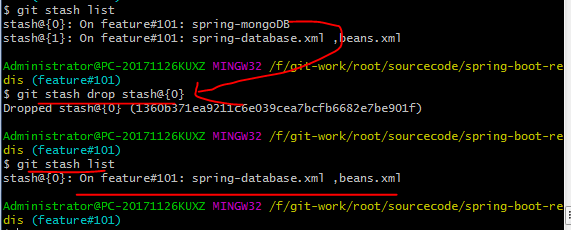
git stash list: 删除所有缓存的stash，等同于：git stash clear

再次在feature#101 上新增一个spring—mongoDB.xml，保存跟新在git stack缓存中：



注意到列表中的stash@{0}、stash@{1}

git stash drop stash@{0}: 删除指定的stash缓存



3,从stash创建分支

这是一个很棒的捷径来恢复储藏的工作然后在新的分支上继续当时的工作。

$git stash branch feature#101changes

