# 一、安装sourcetree

## 1.git插件安装

两个插件：

msysgit是Windows版的Git，从<http://msysgit.github.io/>下载安装即可

Git—preview.exe，未找到官网版本，从网上搜索安装包即可【推荐这个,这个能够在win下使用git命令行】

以上两个插件安装一个即可

## 2.安装sourcetree

安装sourcetree，下载网址：<https://www.sourcetreeapp.com/download/>

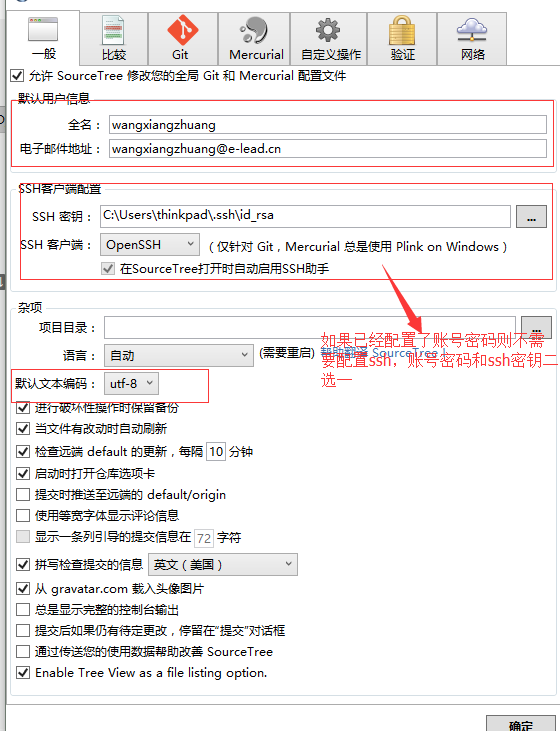
先安装git插件，sourcetree直接按照提示安装即可

## 3.配置sourcetree

打开sourcetree，点击“工具”>“选项”，如下图

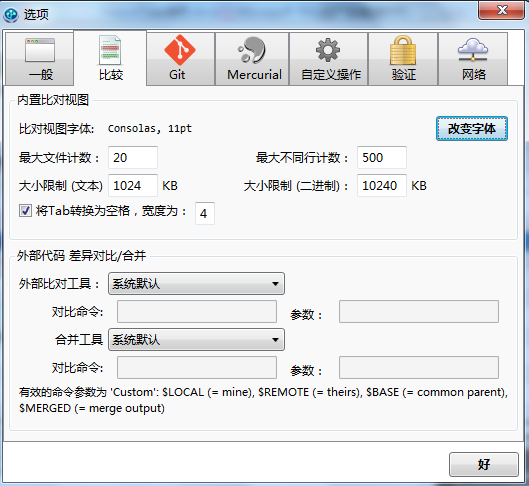
“**一般**”配置页面：

除截图中红圈外，其他均是默认即可



“**比较**”配置页面：

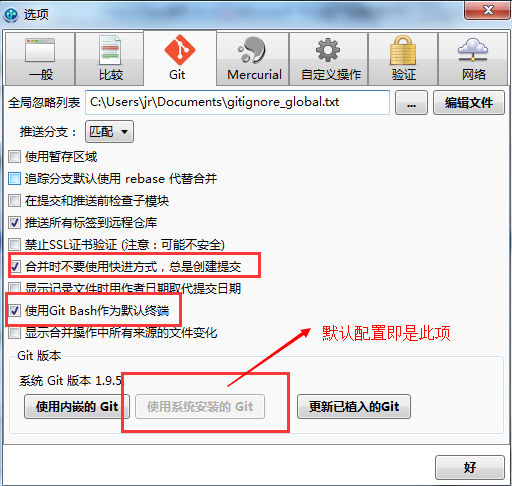
默认配置即可，更好的选择是使用Beyond Compare 对比文本差异



“git”配置项：

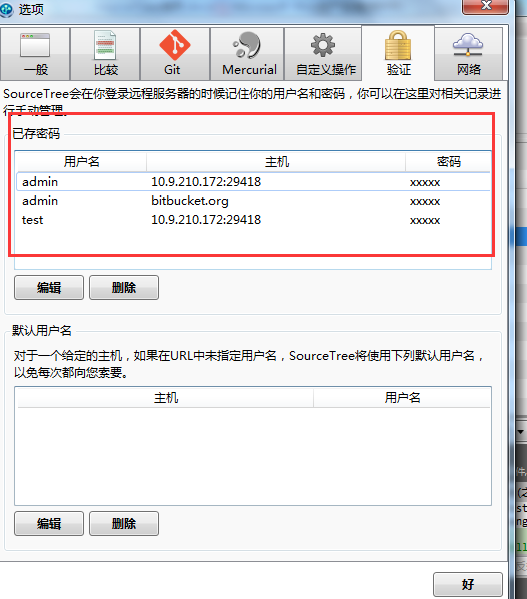
“**合并时不要使用快进配置项，总是创建提交**”---配置项意味着每次分支合并都会创建新的提交节点，推荐这样的使用方式

“**使用git bash 作为默认终端**”–如果同时安装了sysmgit和gitbash时，此配置项生效，建议使用gitbash



“**验证**”配置页面：

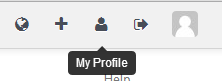
此配置项用于添加远程服务器交互时的身份验证，采用“账号/密码”方式，如果前面已经配置了公钥私钥访问的方式，此配置项可以忽略



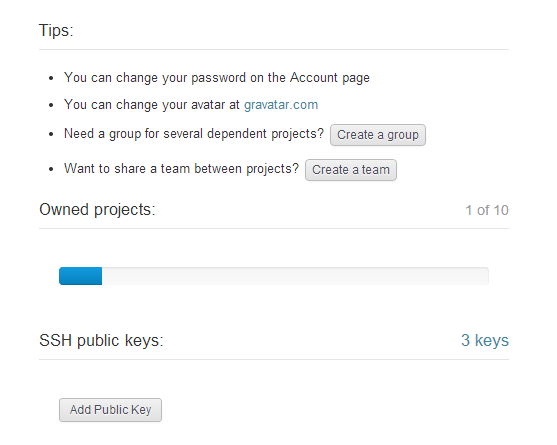
# 二、配置gitlab

添加SSH key

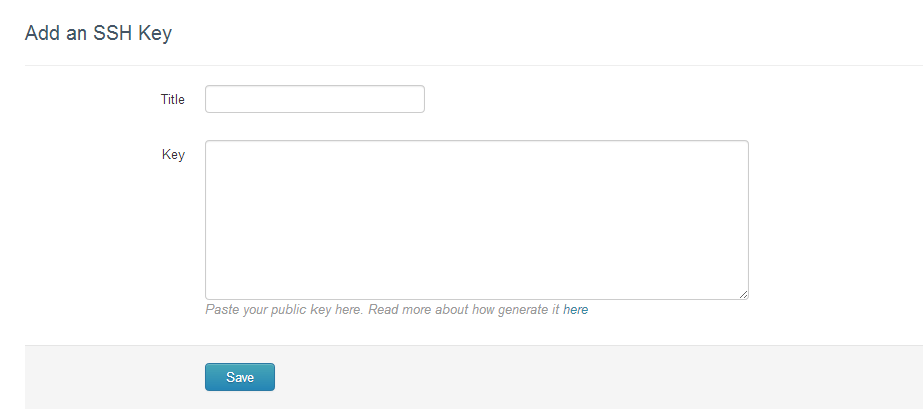
## 1.点击My Profile



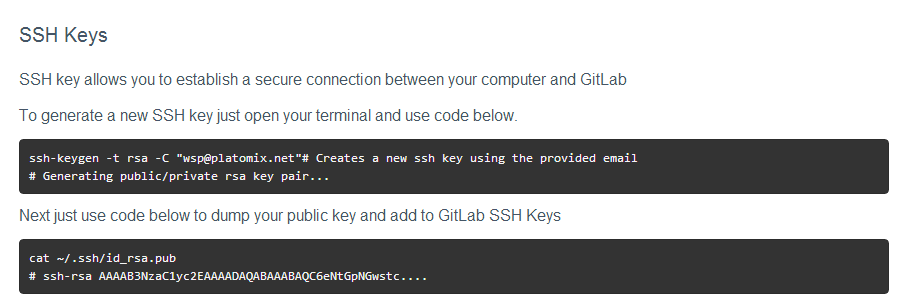
## 2.点击SHH pulibc keys下面的AddPublic Keys



## 3.点击key文本框下面的 here



## 4.出现添加SSH 的示例

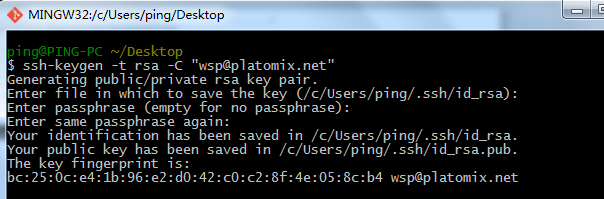


# 三、sourcetree认证配置（秘钥和密码方式）

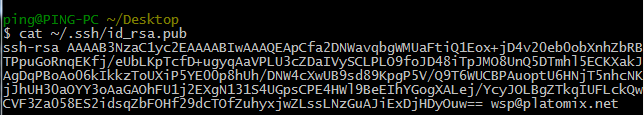
## 1在桌面上打开GIT BASH 终端



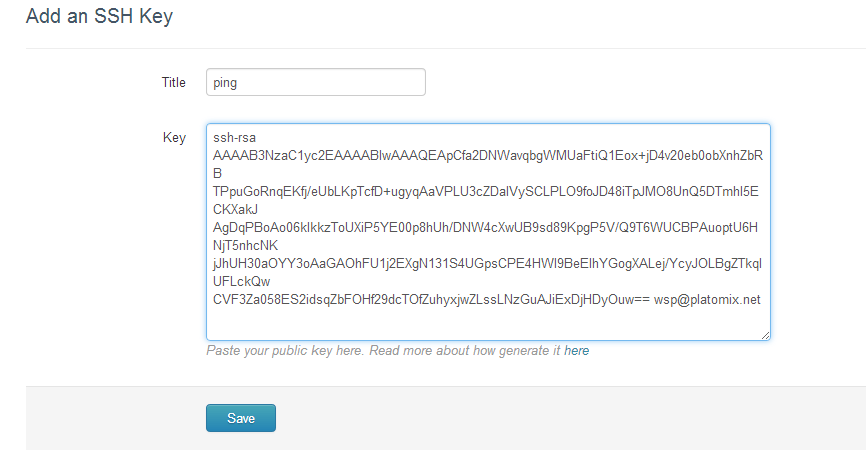
## 2.复制第一行中的复制到终端中，然后一直回车



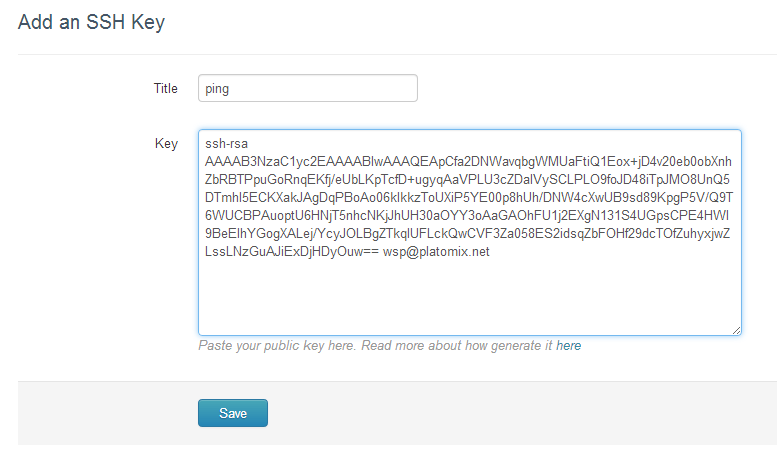
## 3.复制第二行中的到终端中，然后回车



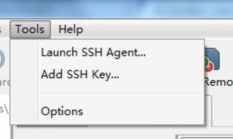
## 4.复制该段key 到gitlab中



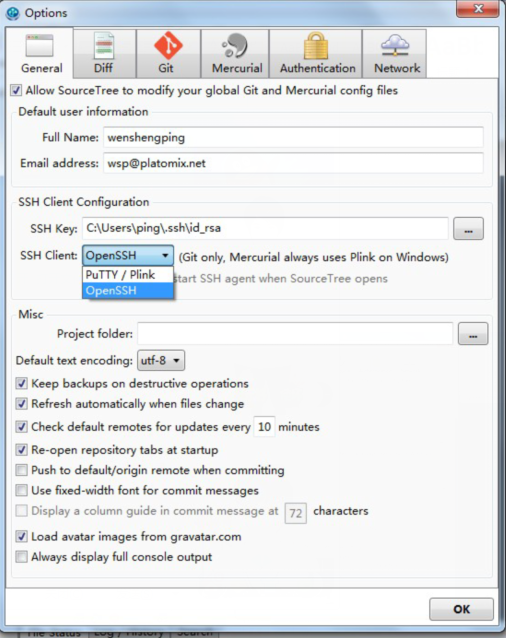
**然后将其中第二段以后的换行缩进，然后点击Save**



## 5.切换到SourceTree,选择Options



**然后在SHH Client选项中选择OpenSSH,其他的不需要修改**



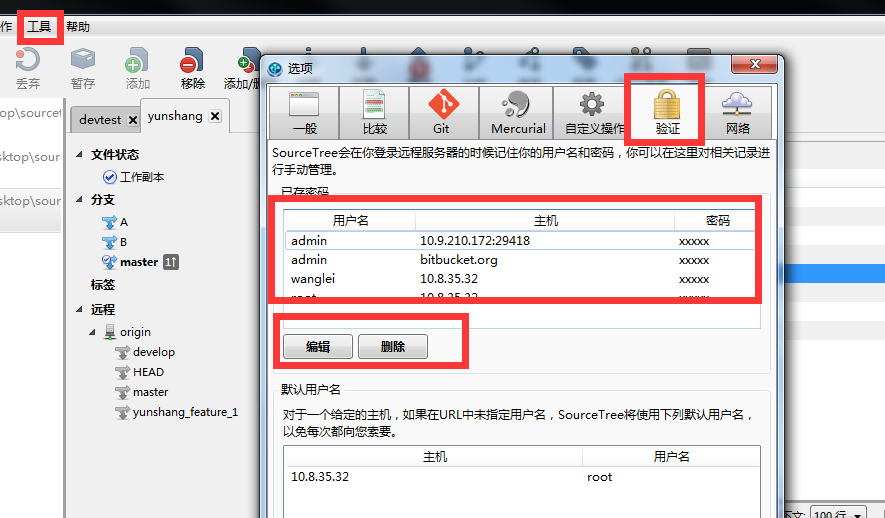
6、账号/密码登陆

每次克隆仓库时，sourcetree会要求输入账号密码，录入后会自动保存账号和密码，

保存位置：工具->选项->验证，如下图

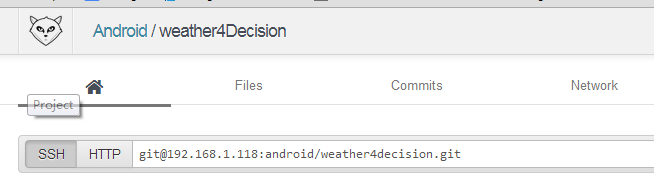
如果要求改，点击编辑，更新密码

如果不需要，直接点击删除

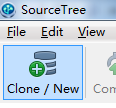


# 四、开发使用场景

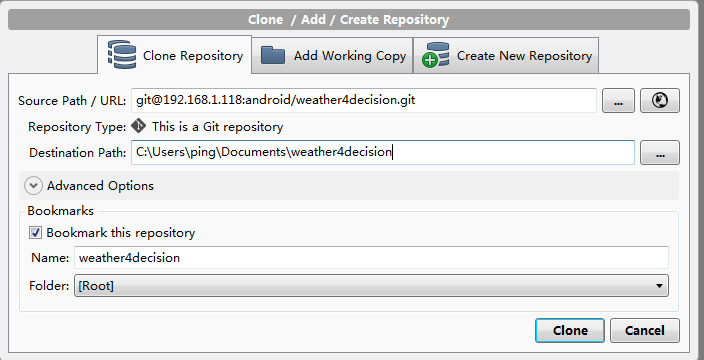
## 1.在gitlab中选择一个项目，复制地址，然后clone到SourceTree中



## 2.点击Clone



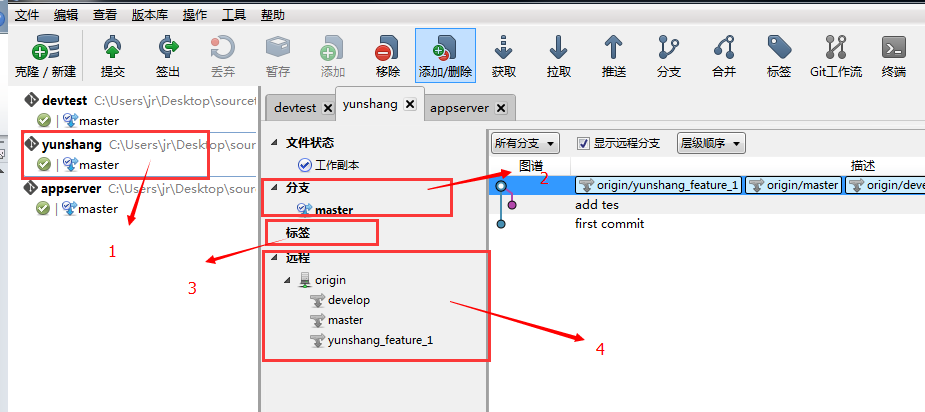
## 3.将复制的地址粘贴到Source Path中，然后将鼠标移动到Destination Path 文本框上点击。如果出现this is a Git repository ,表示成功，然后点击Clone. 如果是第一次使用的话，会弹出输入账号密码的框，在上面输入你的邮箱和密码就可以了。



## 4. 生成代码仓库

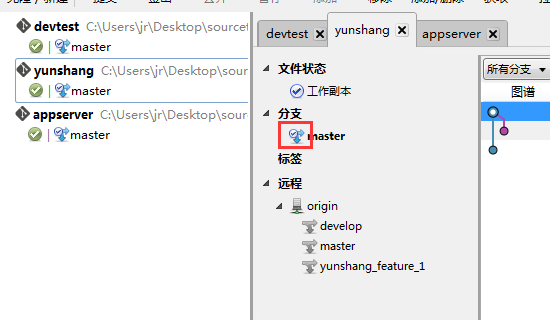
代码clone下来后，如下图所示，

1. 是仓库在本地的标签，不同名称为不同仓库
2. 本地仓库的所有分支：在clone仓库是如果不指定checkout分支，默认本地仓库的分支时master，且只有这一个分支
3. 仓库的标签
4. 远程仓库在本地的副本，每次clone一个新仓库时，会把远程仓库的所有分支都获取到本地仓库，如下图所示有三个仓库：master、develop、yunshang\_feature\_1



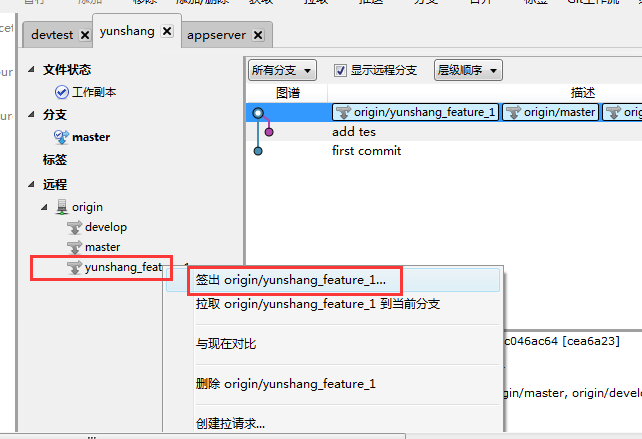
## 5将指定的开发分支获取到本地

首次clone仓库，默认本地分支时master，如下图，master前有个小的打钩“√”，意味着当前的本地分支是master



如果本次模块开发在yunshang\_feature\_1上，则需要将yunshang\_feature\_1 从本地检出

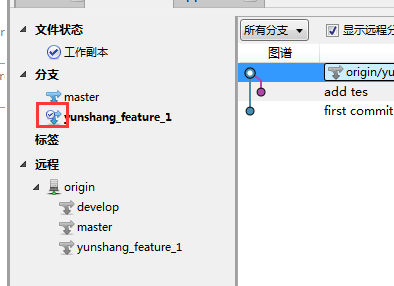
，将鼠标放置远程的yunshang\_feature\_1上，右键，点“签出origin/yunshang\_feature\_1……”



提示如下图，点击“好”



等待数秒，查看当前已是yunshang\_feature\_1分支，

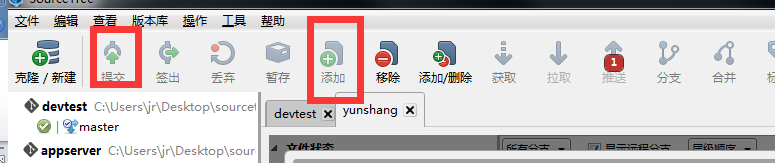


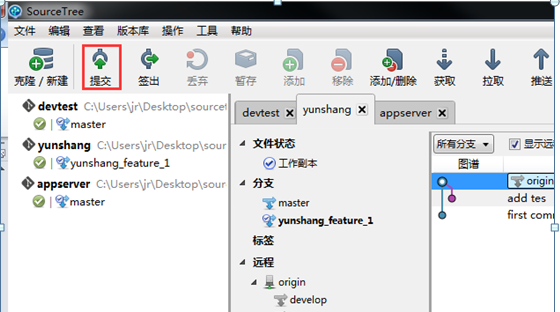
## 6编写代码

开发人员将使用自己的开发工具打开仓库所在文件夹，进行代码开发，在此不做说明

## 7提交版本库

修改文件后，点击左上角“提交”,如果是首次创建的文件，需要在先点击“添加”按钮



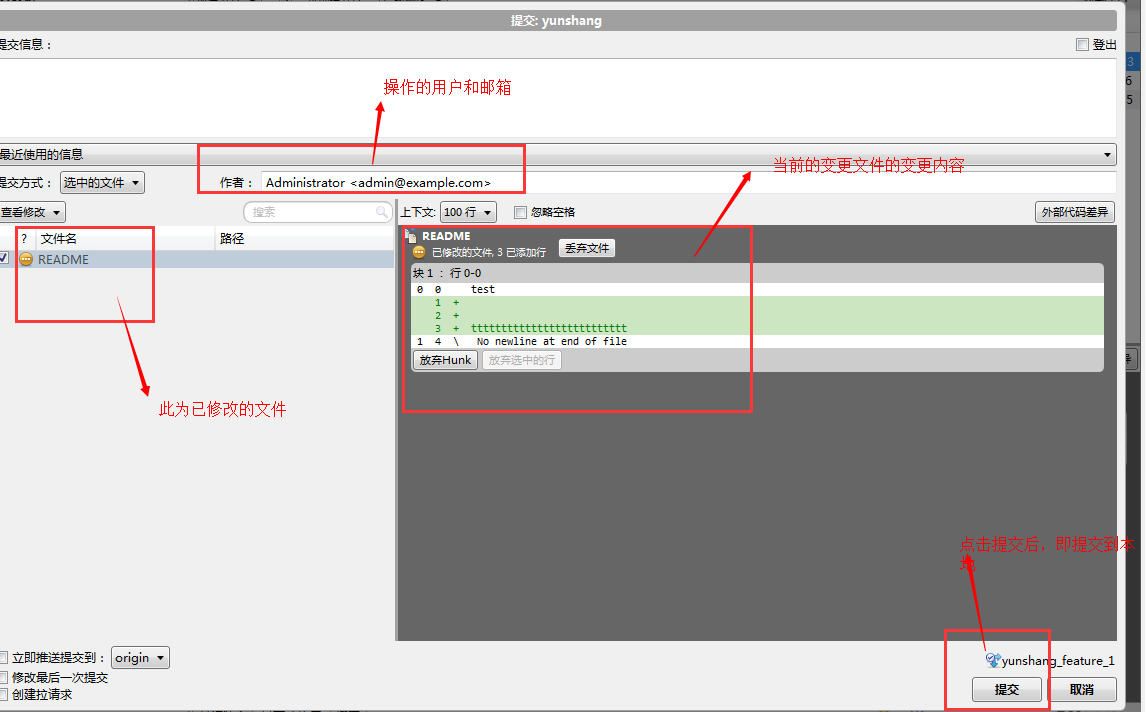


会弹出提交页面：

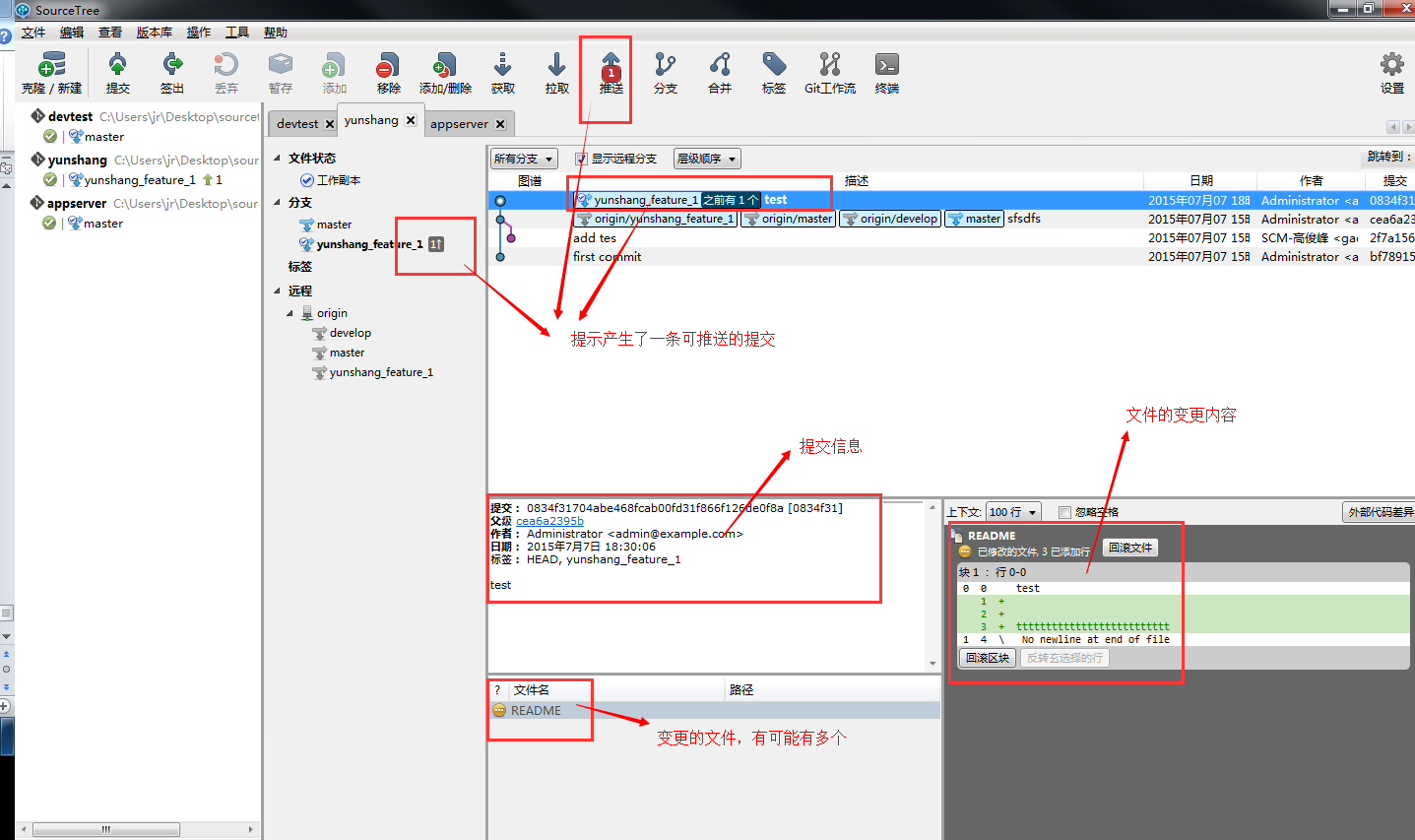
在提交信息一栏，填写提交信息，不填写提交信息会有提示，要求每次提交必须填写提交信息

核对每个文件的变更内容

如果无误，点击右下角进行提交

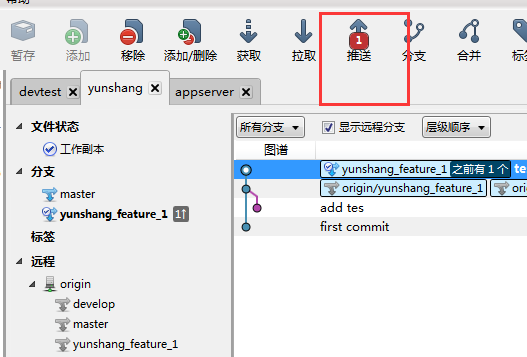


提交成功，会显示如下：



## 8、自测通过，推送至远程yunshang\_feature\_1分支

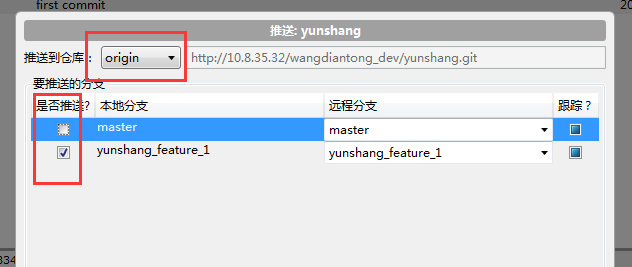
点击“推送”



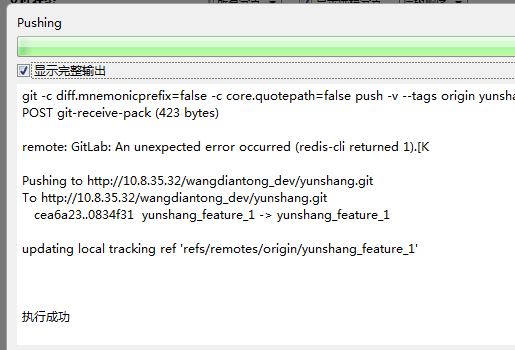
弹出推送页面：

勾选yunshang\_feature\_1，则推送至远程的yunshang\_feature\_1

注意：要求推送时，本地和远程名称必须一致，也就是说不允许本地分支随意推送至不同名的远程分支；如果有推送的需求，先在本地建立相同名称本地分支，再推送至同名远程分支



推送成功：



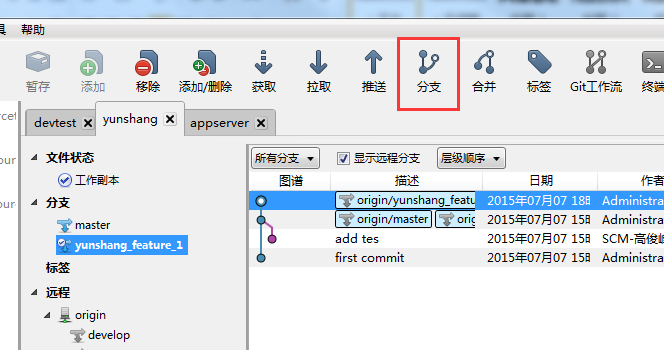
## 9、创建本地分支

情景：开发如果想更自由更方便的做一些开发或测试相关的代码，不想污染了比较正式的开发分支，则可以在本地继续创建分支。

本实例中，本地yunshang\_feature\_1是自己标准的本地开发分支，不想过度提交或者其他原因，则可以在本地yunshang\_feature\_1创建新的分支

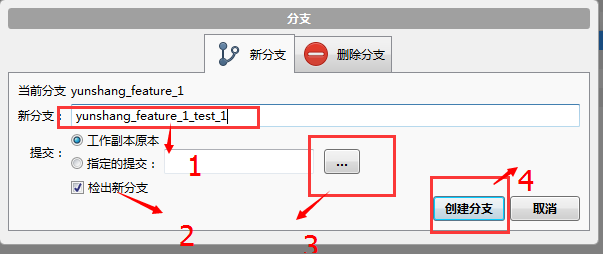
操作：

选中yunshang\_feature\_1，点击“分支”按钮

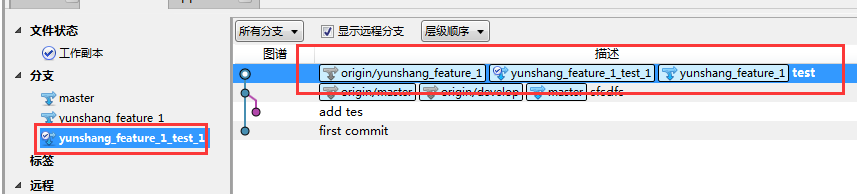


弹出框如下：

1. 新的分支名称
2. 创建后直接切换到这个分支
3. 可以选择在这个分支的指定版本创建新的分支，如果不指定某个提交，则默认从最新的提交创建分支
4. 点击按钮创建



分支创建成功，通过途中红框可以看出，多个分支的最新提交版本一致



现在开发可以在yunshang\_feature\_1\_test\_1上随意提交任何东西，既不会影响别人，也不会影响自己，即使代码污染了，也可以右键直接删掉此分支

## 10、分支合并

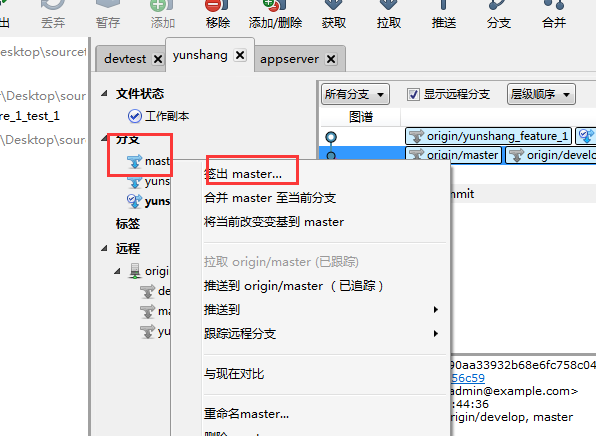
假设开发在本地yunshang\_feature\_1上开发了新功能，但是本次需要提交到master上

步骤如下：

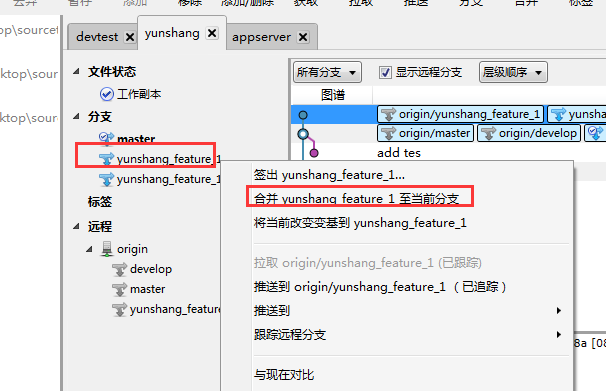
1. Yunshang\_feature\_1合并到本地master
2. 更新远程master仓库，并合并到本地master
3. 本地master分支推送到远程master分支

操作如下

1. Yunshang\_feature\_1合并到本地master：首先切换到本地master



本地yunshang\_feature\_1合并到本地master



2更新远程master仓库，并合并到本地master

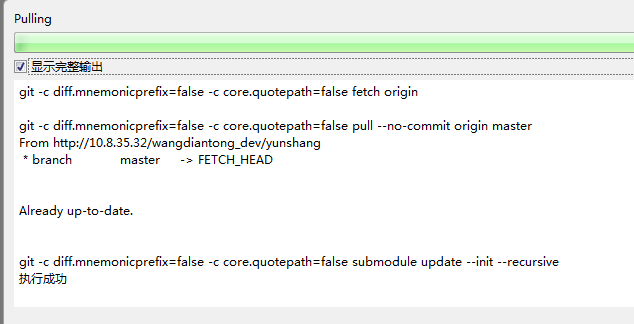
选中master，点击右键，点击红框按钮，



选择“好”

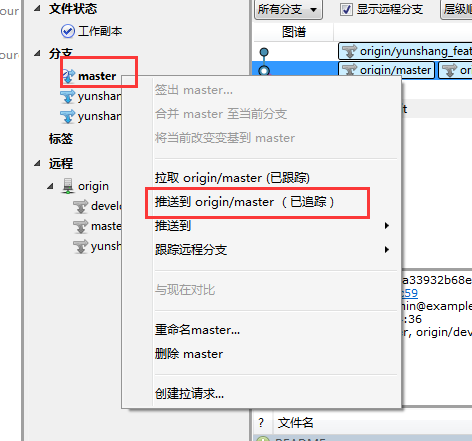


提示合并成功



3、本地master分支推送到远程master分支

点击“推送到origin/master（已追踪）”





提示推送成功

## 11、冲突解决

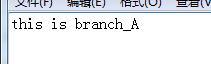
分支合并时会产生冲突，解决冲突办法和svn一致：

分支A和分支B都是从master同一次提交创建而来；

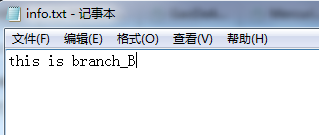
冲突解决方法：

分支A合并到分支B，产生冲突，找出冲突文件，将A中冲突内容改成和B一致（或者将A冲突部分置为空，再提交，即只要保证内容不冲突）后，将修改内容再次提交到A中，然后将A合并到分支B

此时分支A的info文件内容是：

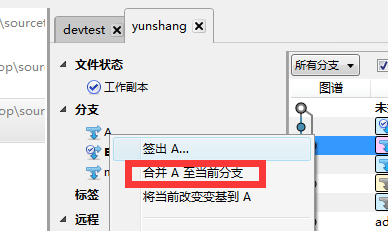


分支B的info文件内容

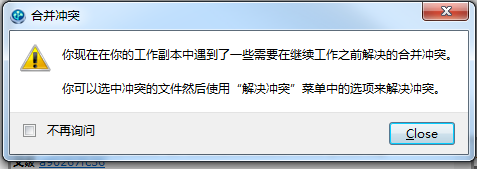


合并操作：

1. 切换版本库至分支B
2. 合并分支A到B

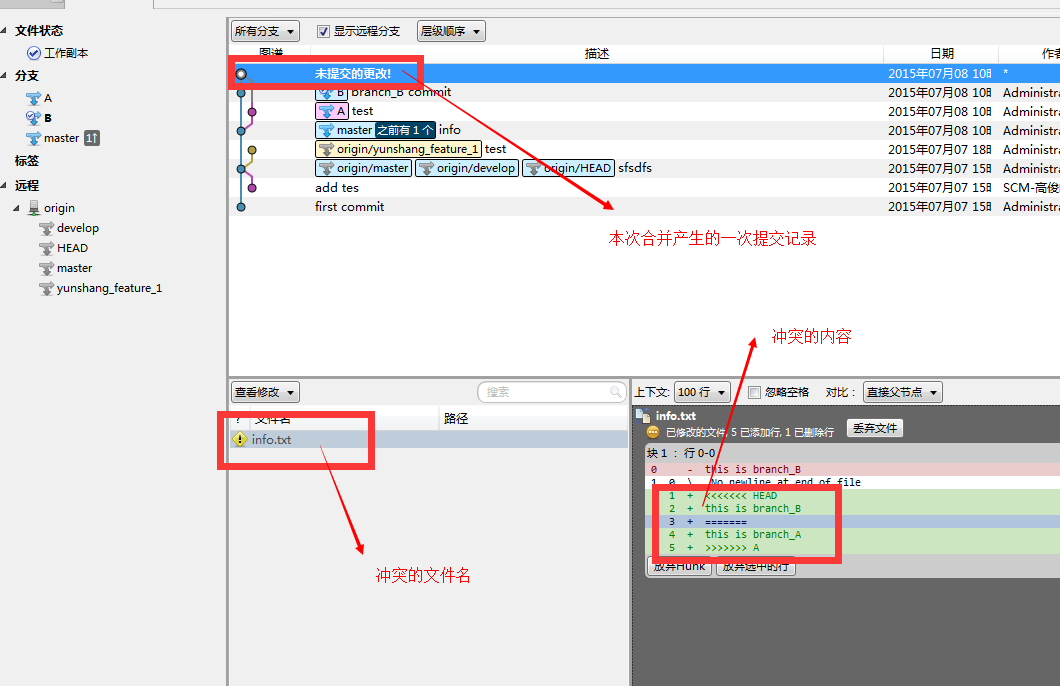


会弹出冲突提示：

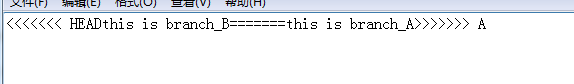


冲突的内容提示：

1. 本次合并产生一条新的提交记录
2. 显示出本次合并产生的冲突文件，合并量大时可能不止一个
3. 显示当前冲突文件的冲突内容，途中显示的冲突内容与上文分支内容一致



打开冲突文件查看，和sourcetree界面显示一致：

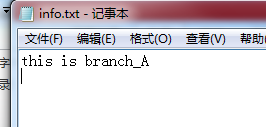


1. 冲突解决方法

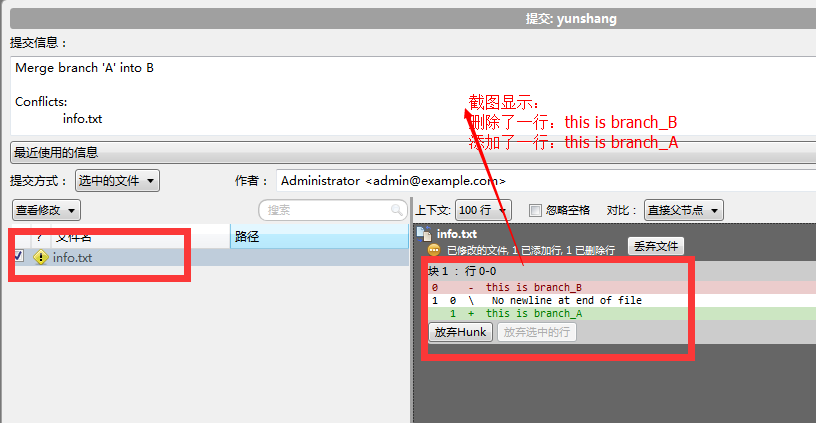
目前看，当前的检出版本是分支B（注意，由于冲突的存在，检出的分支已经不可切换了，不信可以尝试下）

将B中与A冲突的文件内容，改成和A一致；去掉显示的”<<<<<<<<<<<<”、“=========”和”>>>>>>>>>”等提示符，再在分支B中重新提交。

B分支修改后info文件：



点击“提交”



合并成功

