# Git Sourcetree视图工具

作者：孙元清

前言：

本教程已经包含常用的git功能，如果觉得有帮助到在开发中的你，请给这个项目点赞，后期使用工具报错等问题统统可以问我^\_^，我已踩坑！现在觉得是如此好用！！！.

我的微信：SunYuanQing8

1. 它是一个怎样的工具?
2. 它是和git深度融合的一个git版本管理的视图工具,当然它和终端命令行没有什么区别,一个是视图,一个是命令行而已.

它和终端执行的命令是同步的一体的.

1.界面工作区面板介绍

小提示它要账号登录,所以拿笔记好,怕以后忘了.

账号登录的平台在国外...

设置git仓库的账号的用户

查看当前用户名和邮箱

git config user.name

git config user.email

修改当前用户名和邮箱

git config --global user.name "你的名字"

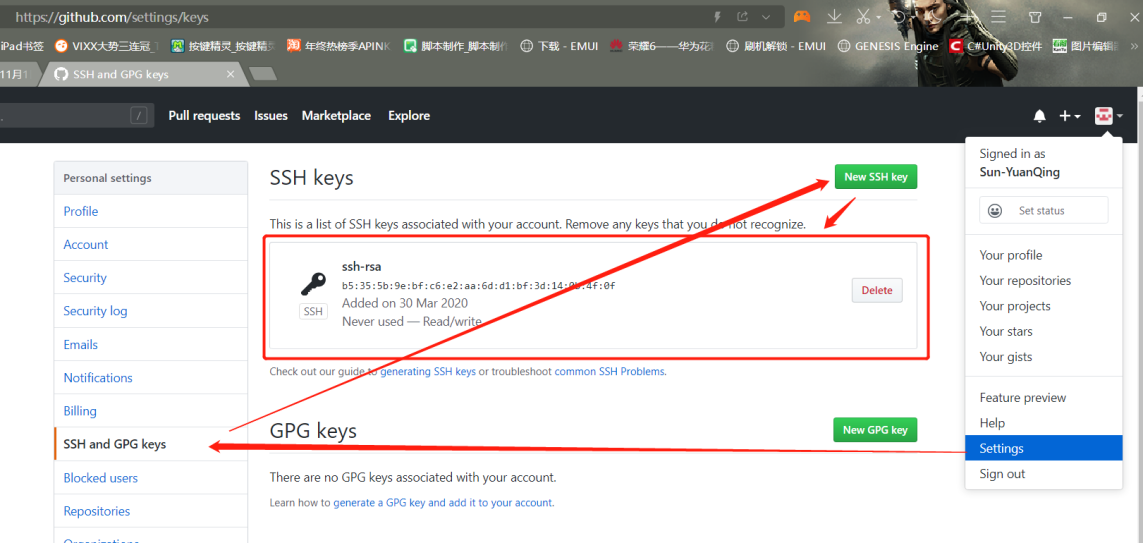
git config --global user.email "你的邮箱"

~~设置公钥ssh-keygen -t rsa -C "你的邮箱"~~

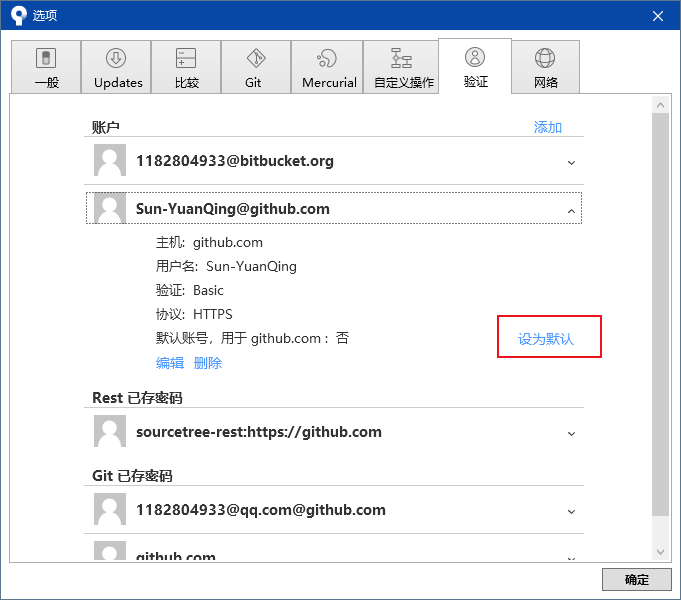
然后输入生产ssh key的命令：

ssh-keygen~~或ssh-keygen -t rsa -C "你的邮箱"~~

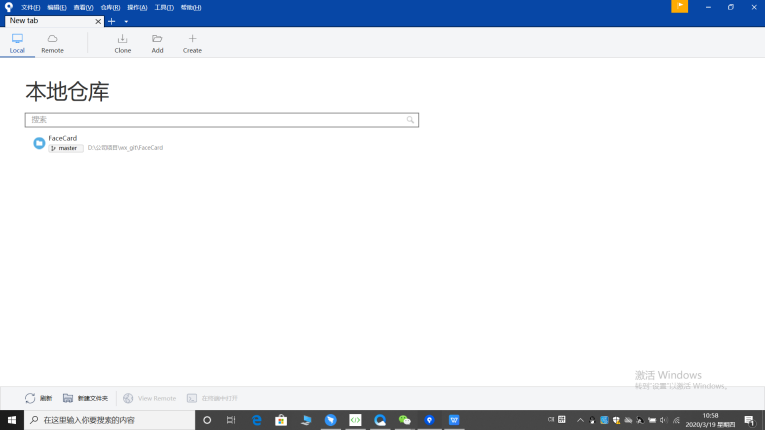
笔记本打开c/Users/lingh/.ssh/id\_rsa.pub.复制内容到GitHub



工具-->选项



1.1安装Sourcetree并设置用户后初始界面是这样的(我已经有了一个包)

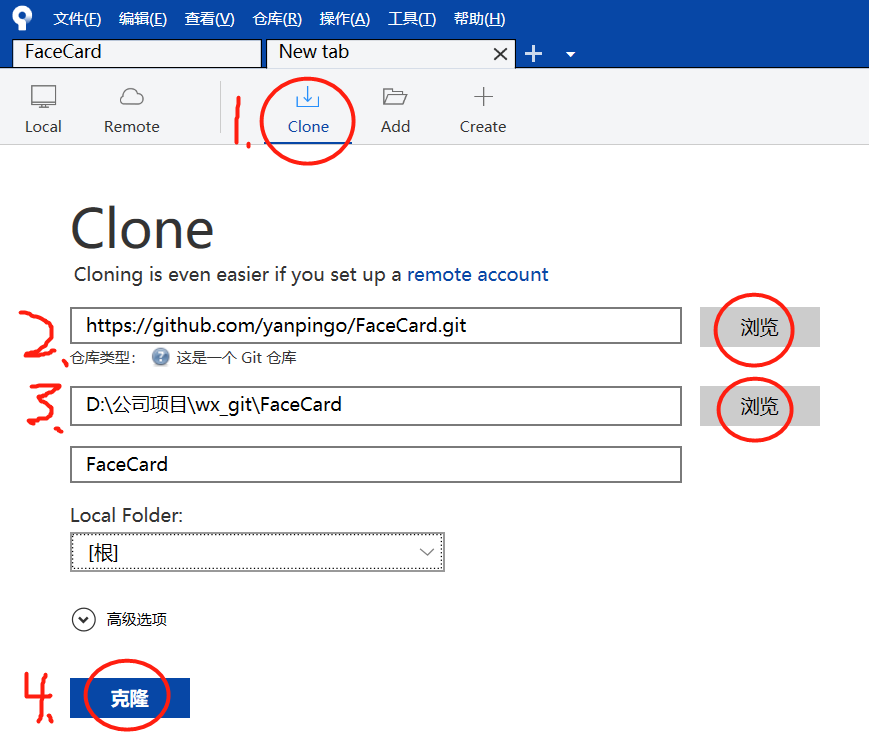


2从gitgithub上克隆一个项目共同开发

我先建一个文件用来放置git包源码

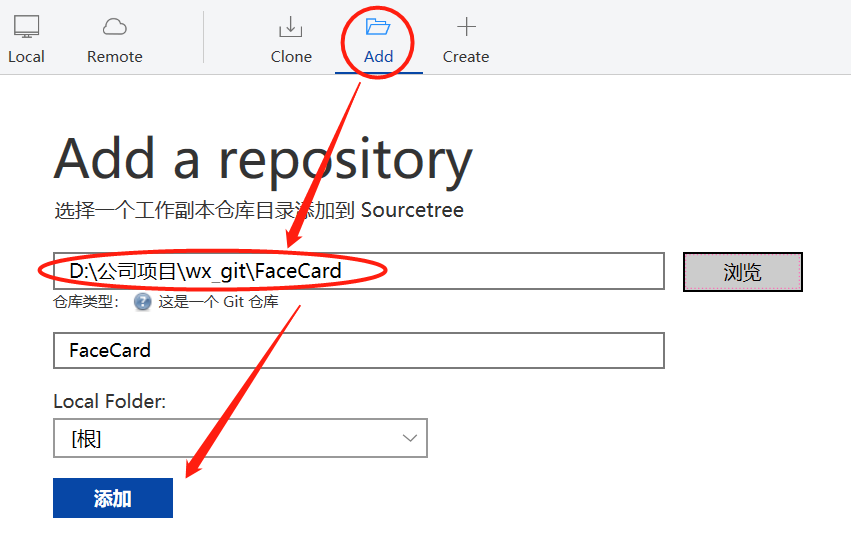


从gitgithub上克隆的目标仓库,并指定放到那里.



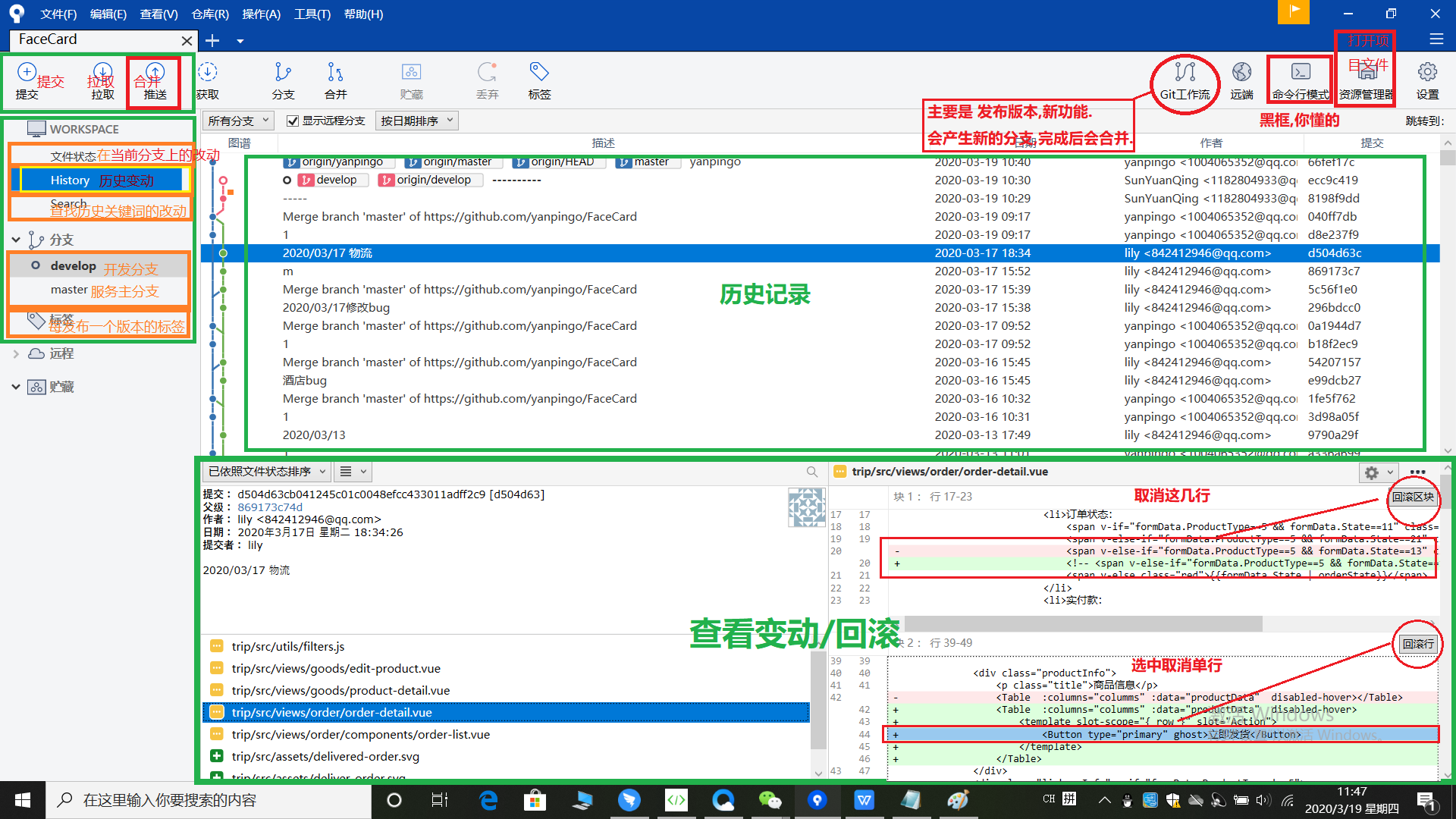
1. 导入现有的本地GIT仓库

由于之前就有了GIT仓库,再用这款软件照样可以同步,选择Add添加本地仓库就好了.



1. 克隆完成,界面工作区的介绍

这个是在develop分支下 查看History项目 历史变动 的记录列表



1. 分支介绍

5.1.Master

它是主干,相当于一颗大树的主干,一般我们不会在这上面直接修改.

所有子分支 新功能的分支,新版本的分支包含develop最终都会合并或推送到Master上,形成一个完整的版本.

5.2.Develop

0.如果没有Develop可以通过Git工作流初始化产生Develop.

1.它是我们开发时的一个分支,最新的代码在这里,这是约定俗成的,新同志可以拉取Develop上开发

2.可能还有基于它的新分支

2.1.新功能(每个人可能1个分支,完成新功能后合并. 后面会介绍 新功能分支 的操作)

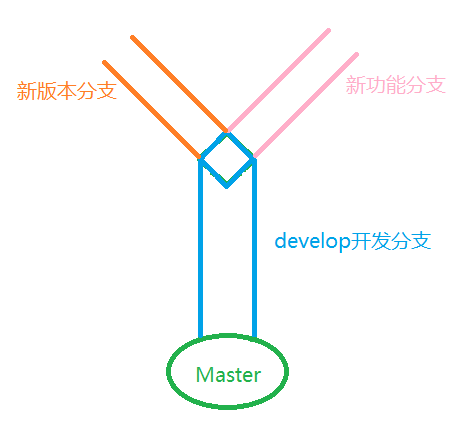
2.2.新版本(版本发布者操作的分支,如果你有幸看到,说明发布者正在发布新的项目版本,随后会合并代码到Develop,Develop再推送到Master. 后面会介绍 发布新版本 的操作)

5.3.其它分支

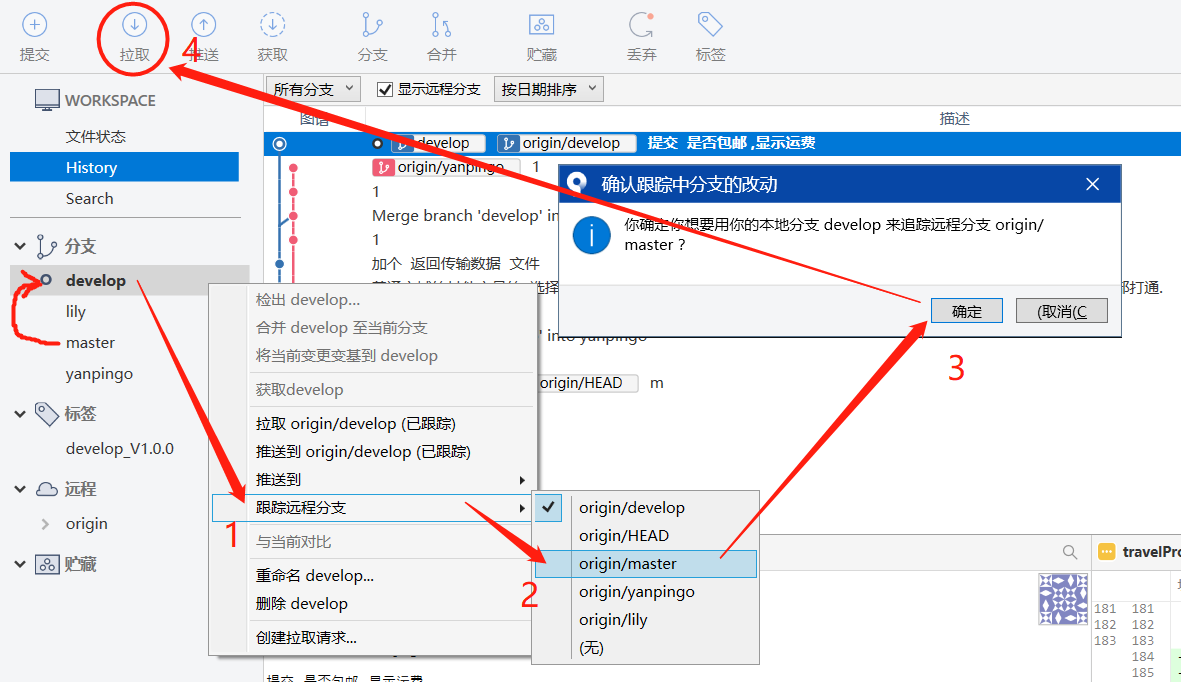


5.4.理想分支(master和develop)

我们的分支,master,develop,yanpingo,lily处在master同一级



5.5.【同步合并】其它分支(Master)代码到本分支

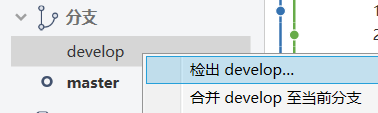


6.简单的 拉取 , 提交

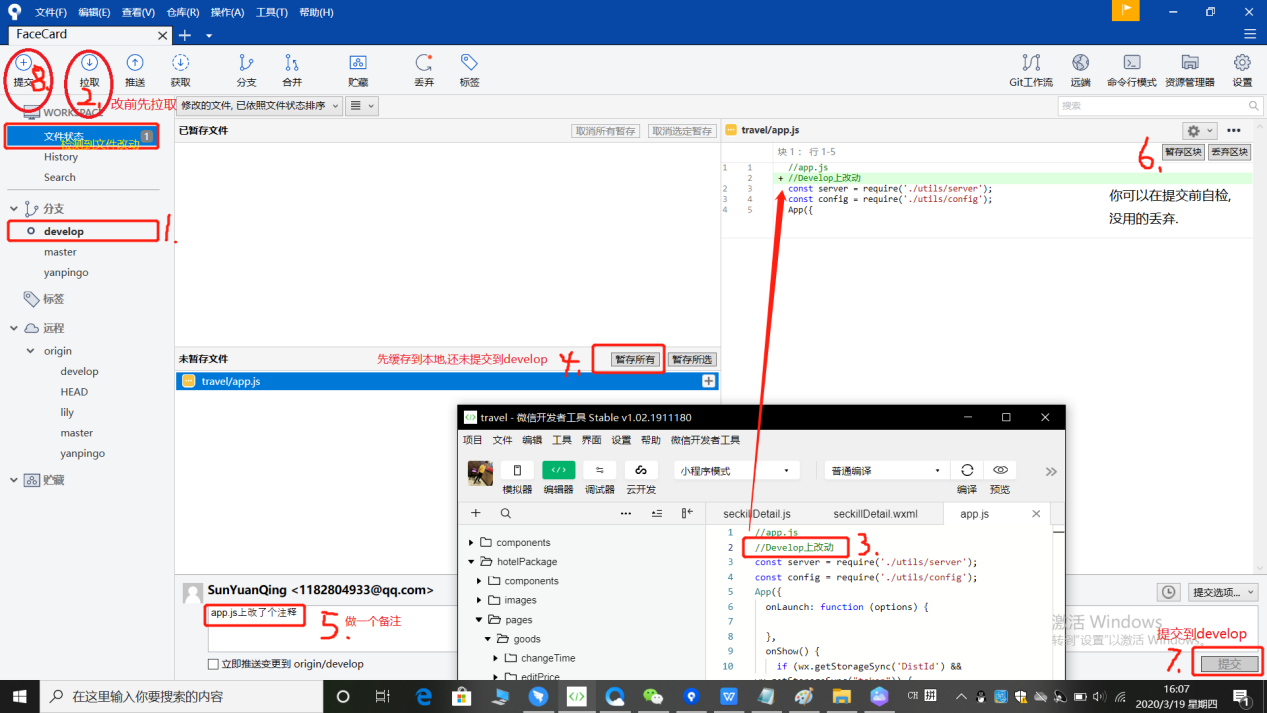
6.1首先我们将分支切换到Develop开发分支上,一切新的改动都基于它.

Develop右击检出develop

注意如果master上有未提交的文件,检出到其他分支会报错,你要么提交要么丢弃更改.

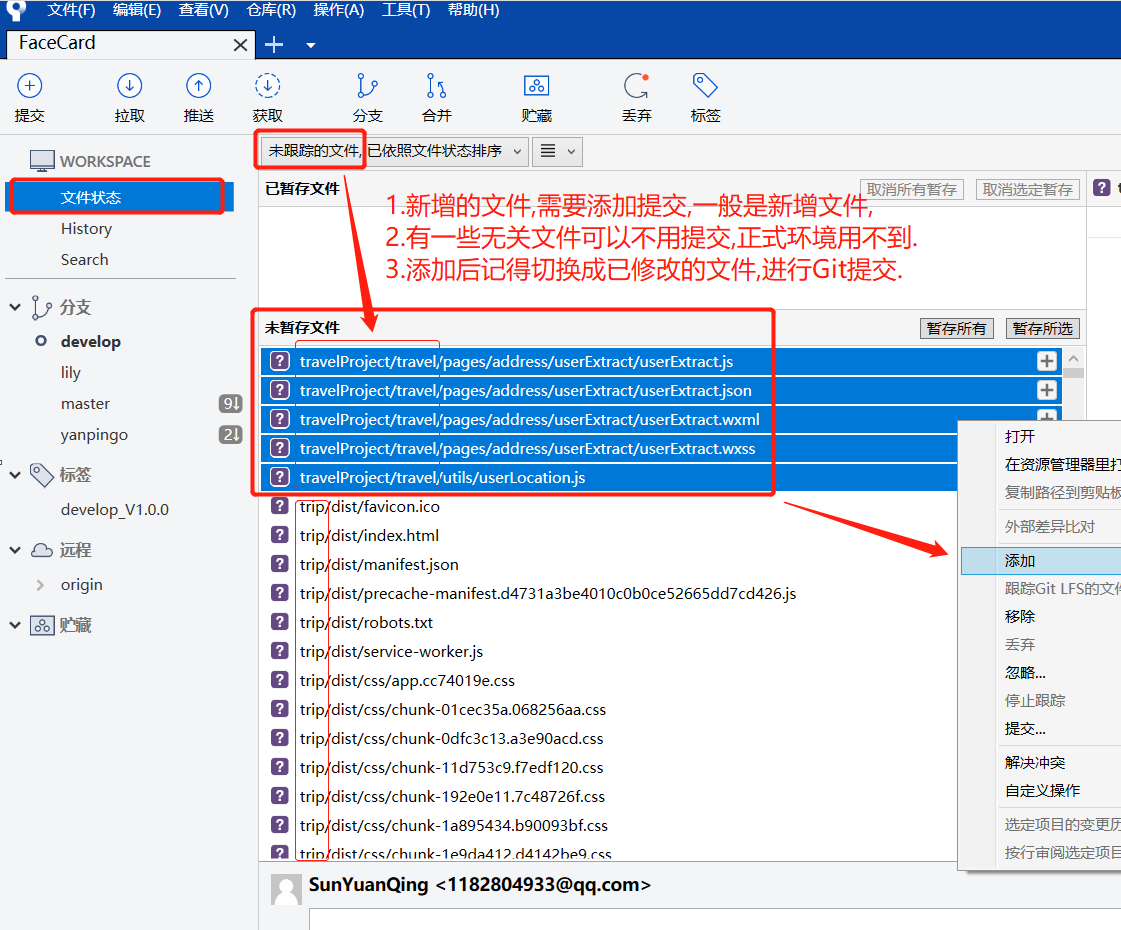


6.2提交,拉取

6.2.1改动之前先拉取

6.2.2新增的文件,需要添加提交

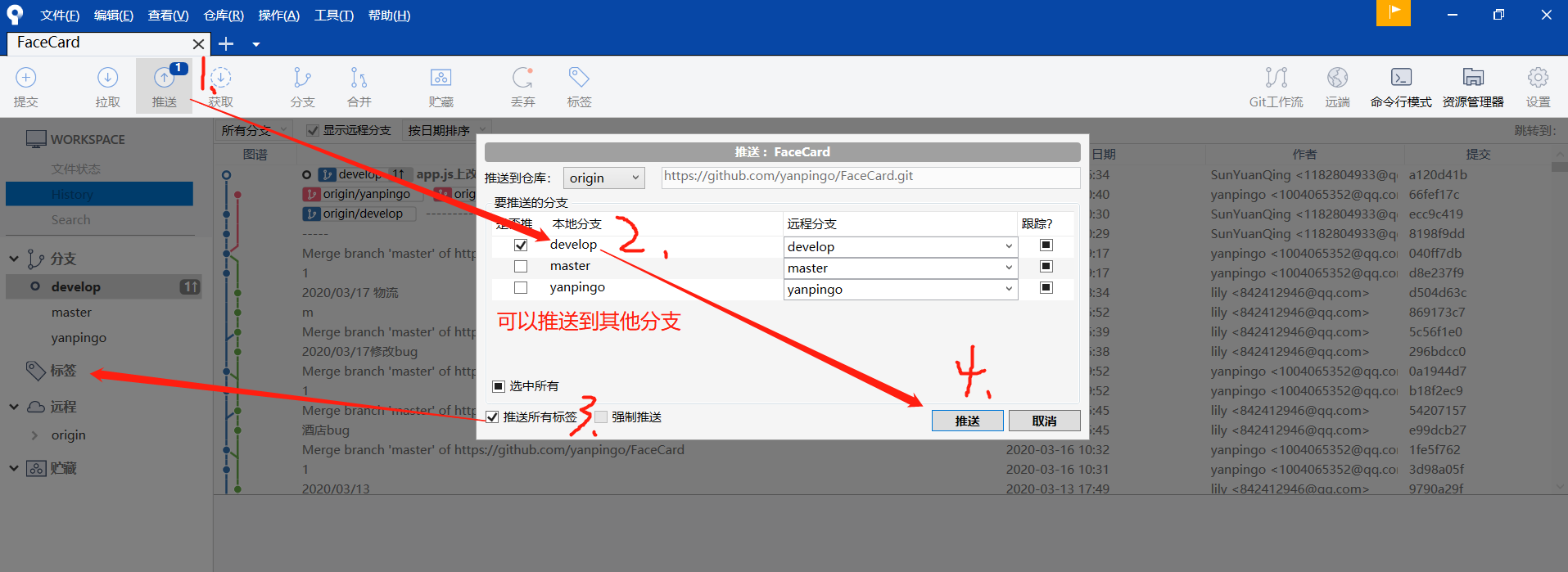
不知道是我设置了还是怎么的,新增的文件,需要添加提交,有一些无关文件可以不用提交.....



6.3推送

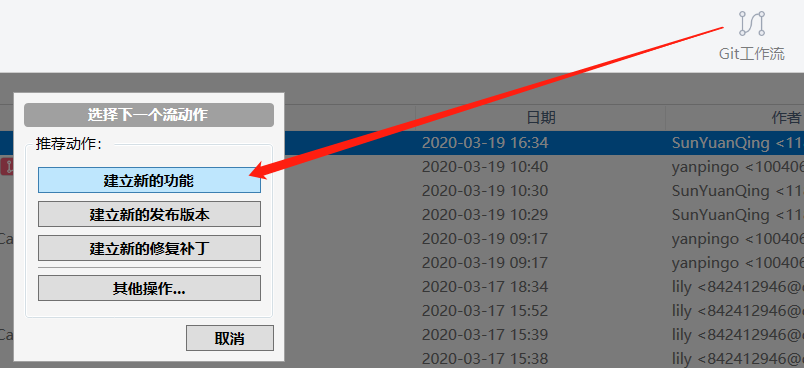
6.3.1推送是将本分支代码给其它分支

6.3.2分支太多了,所以记住先切换到所有的分支都拉取一遍,再推送,不然可能报错,重要,重要,重要!!!!!

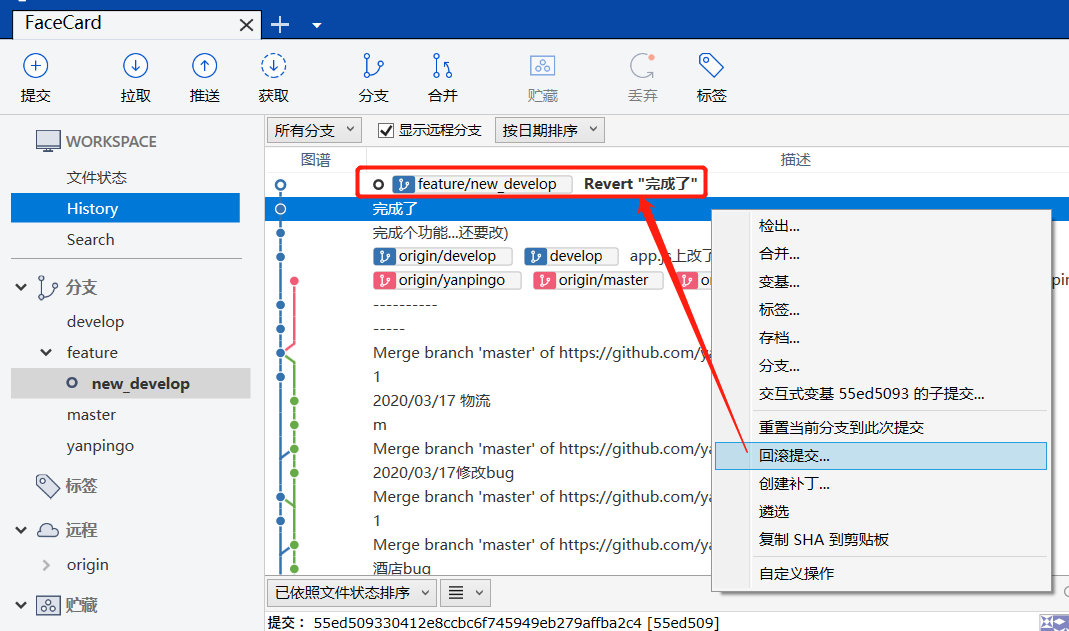


1. [新功能分支]

6.1.Git工作流---->建立新的功能

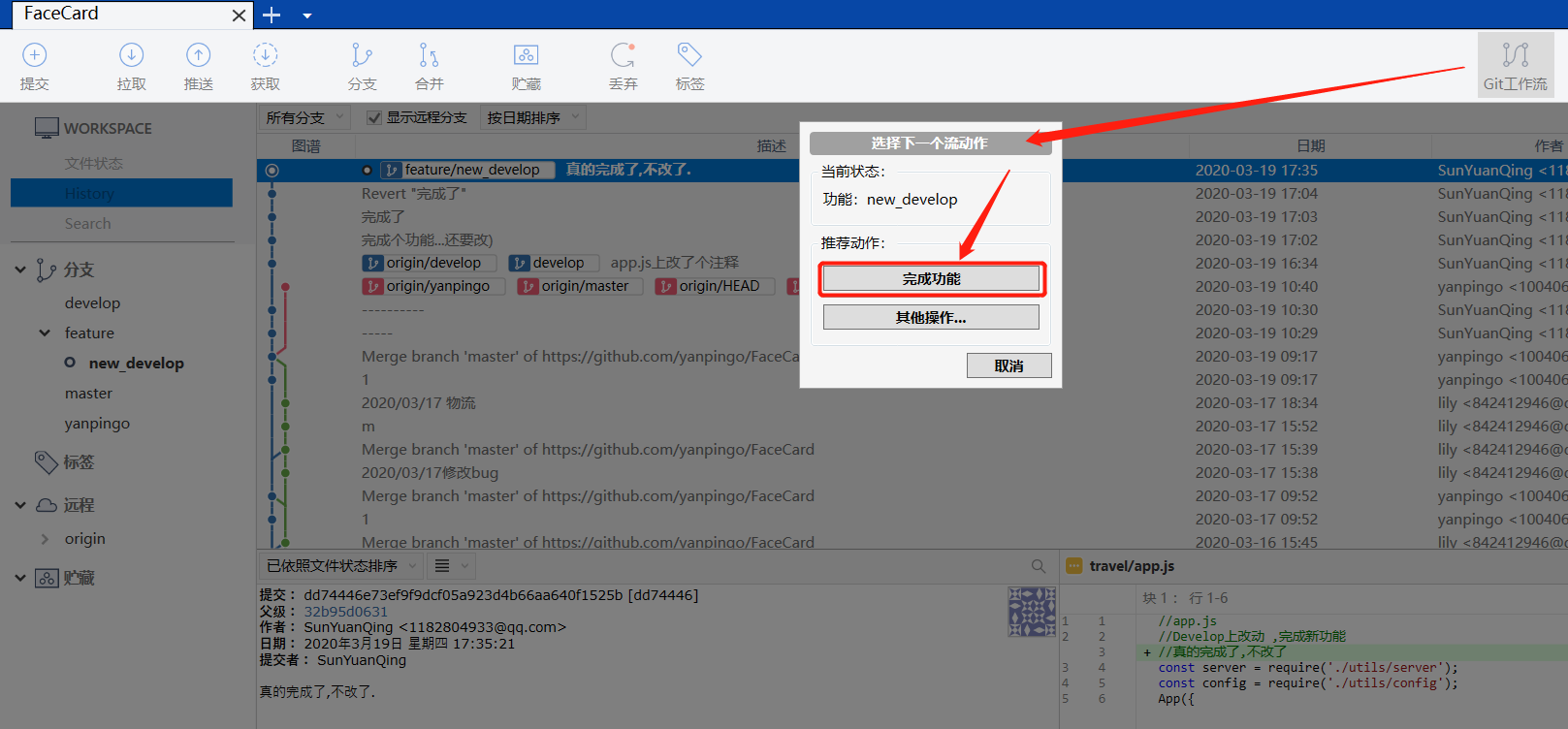


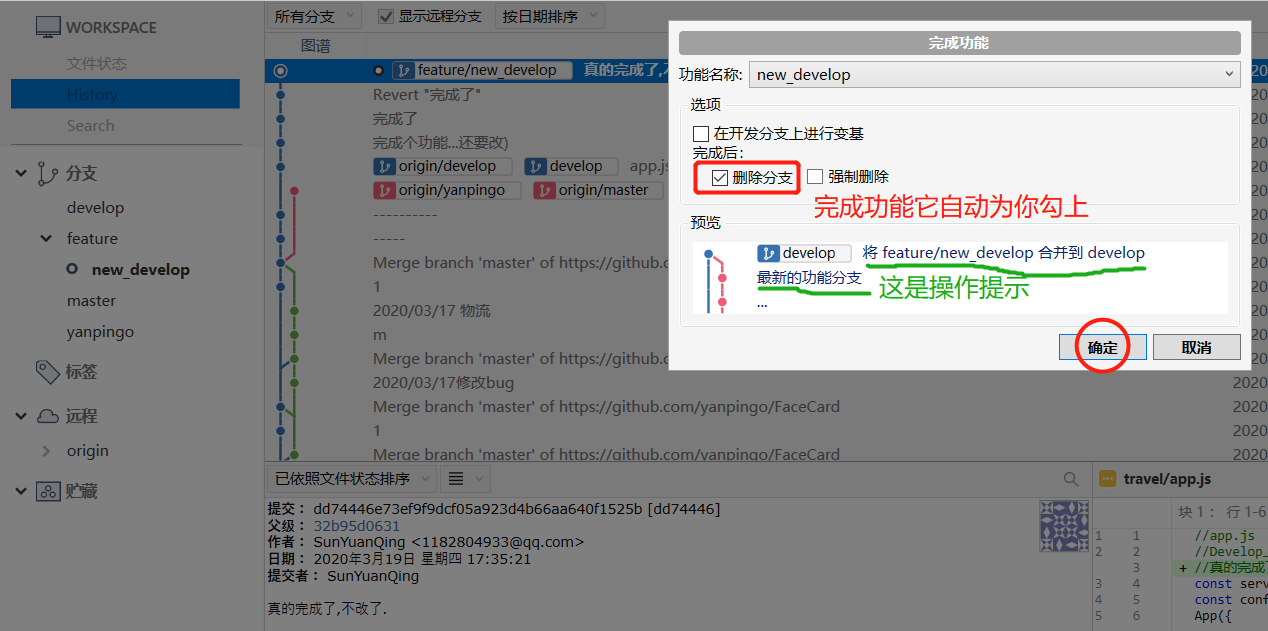


你可以回滚之前提交的代码

6.2.经历过一系列代码改动提交完毕以后,选择

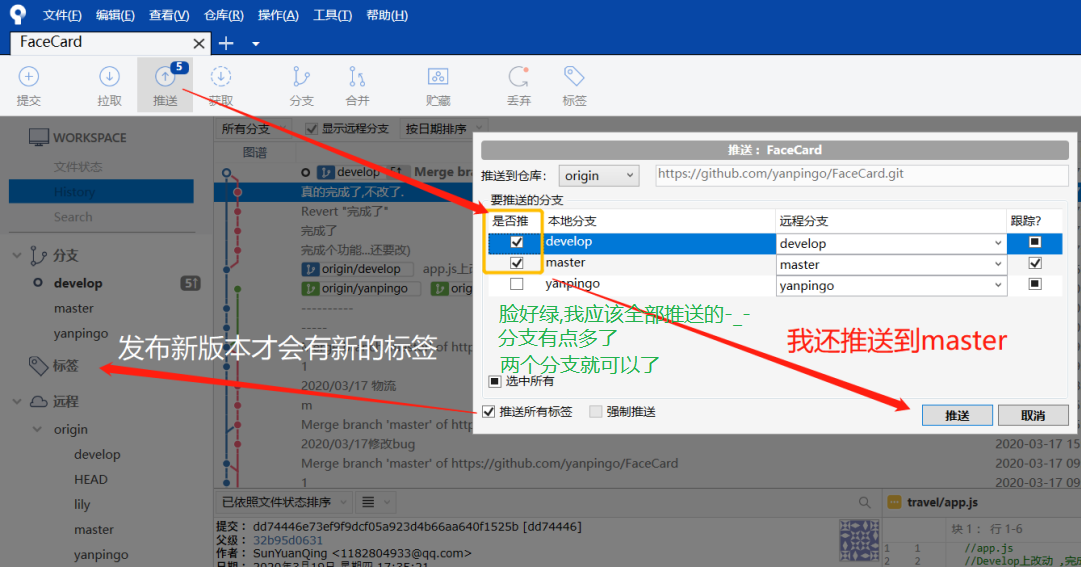
6.3.git工作流--->完成功能





推送是将本分支代码给其它分支,记住先切换到所有的分支都拉取同步合并分支一遍（可能要测试Master先不动，等发布新版本同步合并）,再推送,不然可能报错,重要,重要,重要!!!

6.4再推送

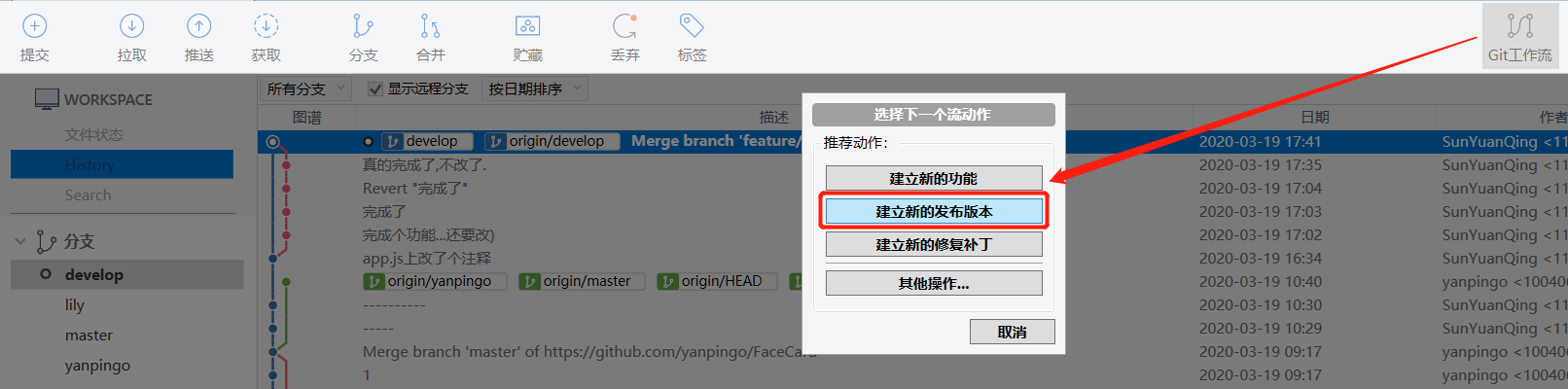




好新功能就介绍完毕了

1. [新版本分支]

7.1.Git工作流--->建立新的发布版本

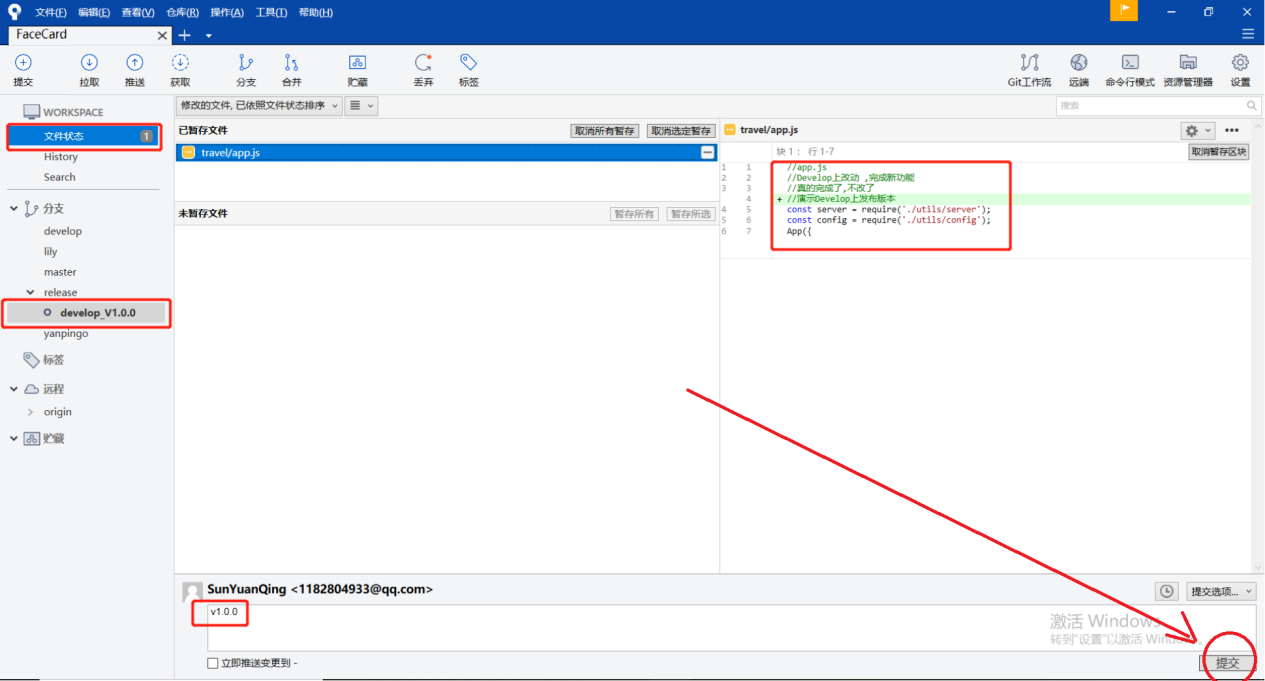


7.2.启一个版本号V1.0.0,确定

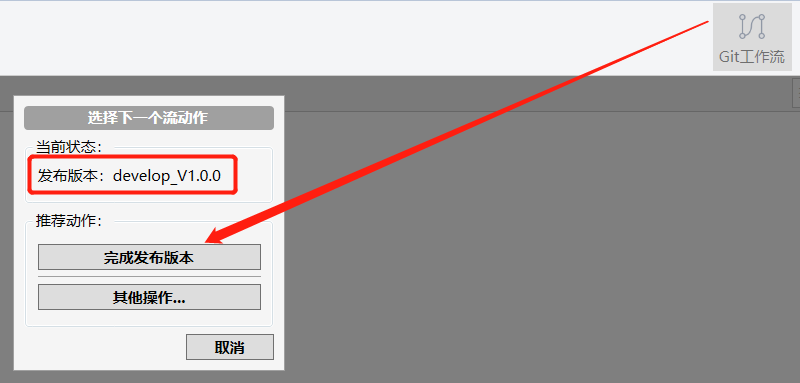
忘了截图...

7.3.改点东西--->暂存所有

7.4提交



7.5.Git工作流--->完成版本发布

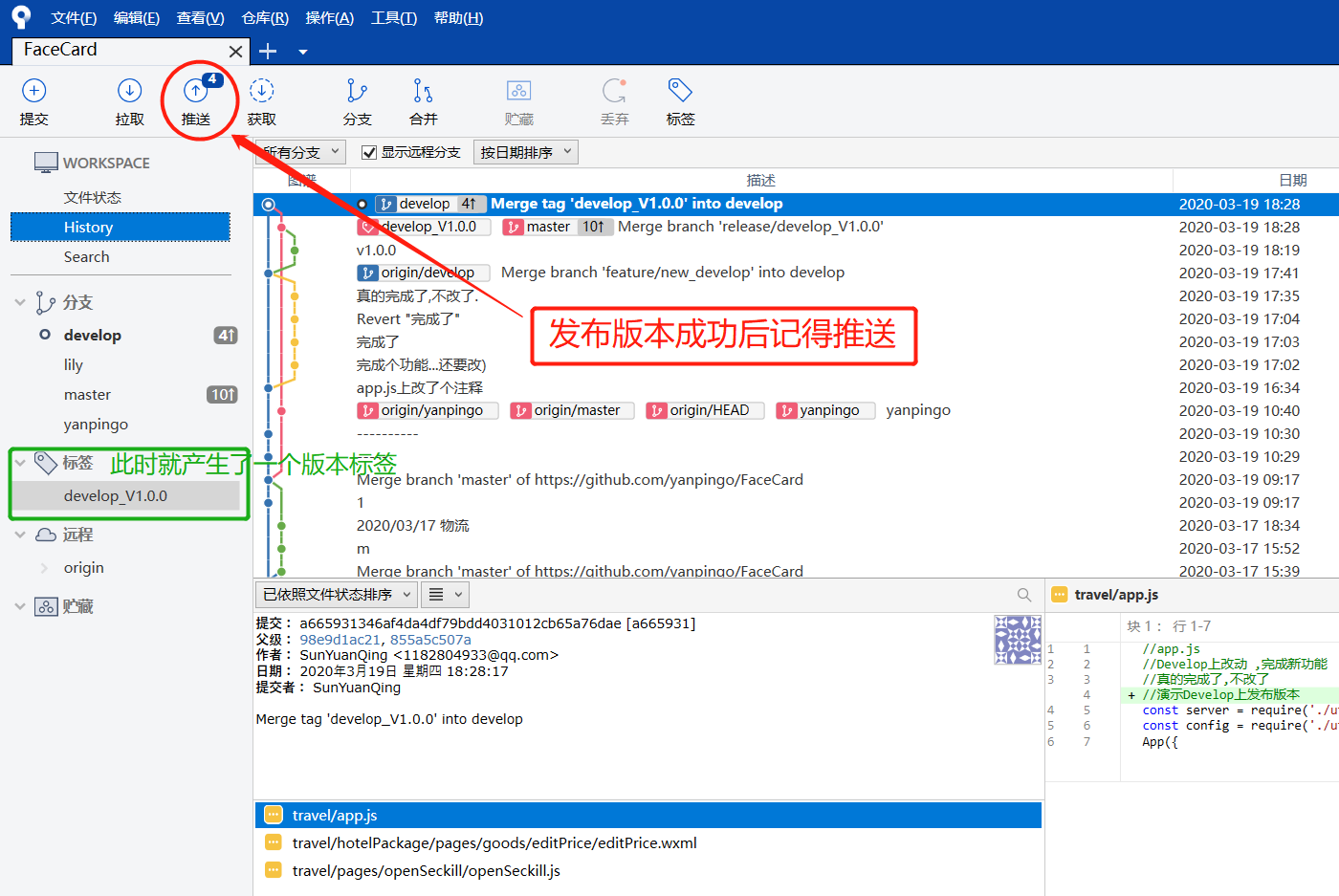




7.6.确定

7.7.分支太多了,所以记住先切换到所有的分支都拉取【同步合并】分支一遍,再推送,不然可能报错,重要,重要,重要!!!!!

7.8.推送



所有分支都

