

git

1、git实操

① 创建版本库

```
git init() --初始化
git config user.name "yrx"
git config user.email "yrx@qq.com"

--global 表示全局属性，所有的git项目都会共用属性（如果不想用全局的就把global去掉）
```

② 提交文件

```
1、查看文件状态：
git status
2、将文件添加到暂存区：
git add 文件名
3、提交文件到本地库（直接带注释提交）
git commit -m "注释内容"
```

③ 查看文件提交记录

```
git log 文件名 --查看历史记录
git log --pretty=oneline 文件名 --简易信息查看
```

④ 回退历史

```
git reset --hard HEAD^ --回退到上一次提交
git reset --hard HEAD~n --回退n次操作
```

⑤ 版本穿越

```
git reflog 文件名 --查看历史记录的版本号
git reset --hard 版本号 --执行可以到这个版本
```

⑥ 还原文件

```
git checkout -- 文件名 --还原文件（在本地将文件盖卵后可通过它来还原）
```

⑦ 删除某个文件

```
先删除文件
再git add 再提交
```

2、分支

① 创建分支

```
git branch <分支名>
git branch -v 查看分支
```

② 切换分支

```
git checkout <分支名>
一步完成: git checkout -b <分支名>
```

③ 合并分支（改变谁，切换到谁上）

```
先切换到主干 git checkout master
git merge <分支名>
```

3、冲突办法

- ① 修改冲突文件（可通过`git diff` 找到冲突的文件及文件内容，再定位进去进行修改）
- ② `git add` 文件名
- ③ `git commit`

4、远程地址

① 增加远程地址

```
git remote add <远端代号> <远端地址>
例: git remote add origin https://github.com/user111/HelloWorld.git
```

② 推送到远程库

```
git push <远端代号> <本地分支名称>。
例: git push origin master
```

注: 之前需要

```
git add
git commit
```

③ 从远程仓库克隆

解释：克隆的默认远端代号为origin

`git clone <远端地址> <新项目目录名>。`

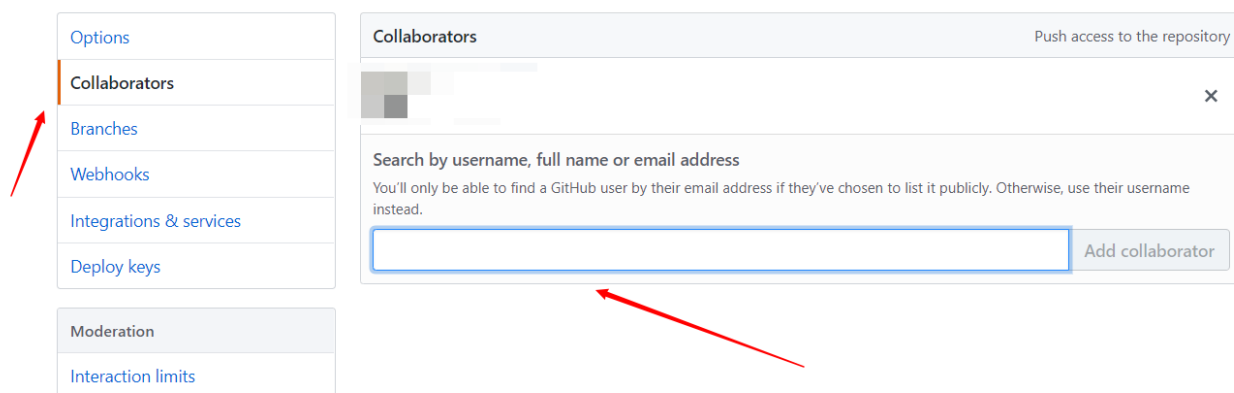
例 `git clone https://github.com/user111/Helloworld.git hello_world`

④ 从远程仓库更新项目

`git pull <远端代号> <远端分支名>。`

例 `git pull origin master`

5、如何邀请别人成为合作者



点击项目右上方的settings后出现如下情况，选择合作者后，git会给它邮箱发送消息，等待同意后即加入小组开发

6、协作冲突

在上传或同步代码时，由于你和他人改了同一文件的同一位置的代码，版本管理软件无法判断究竟以谁为准，就会报告冲突,需要程序员手工解决。

解决冲突办法：

- 1、修改合并
- 2、`git add`
- 3、`git commit`

7、idea的操作

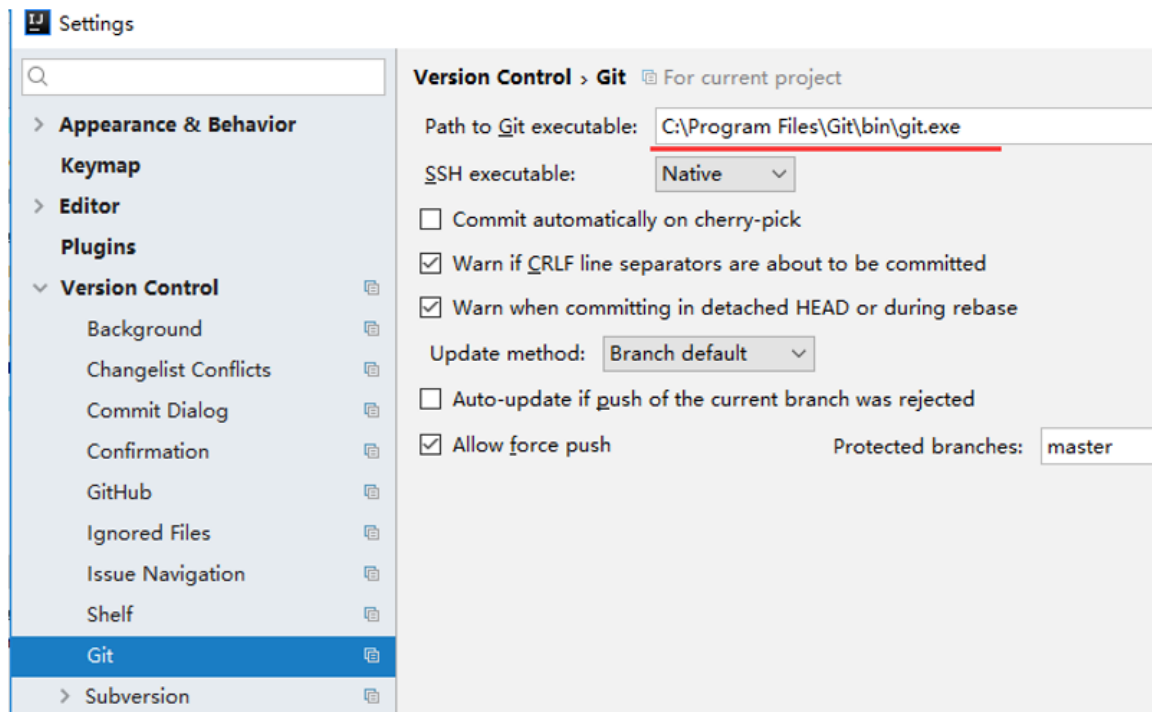
①

在idea中，settings → Version Control → Git

选择本地的Git.exe的位置（替换成自己电脑使用的git）

在idea中，settings → Version Control → Git

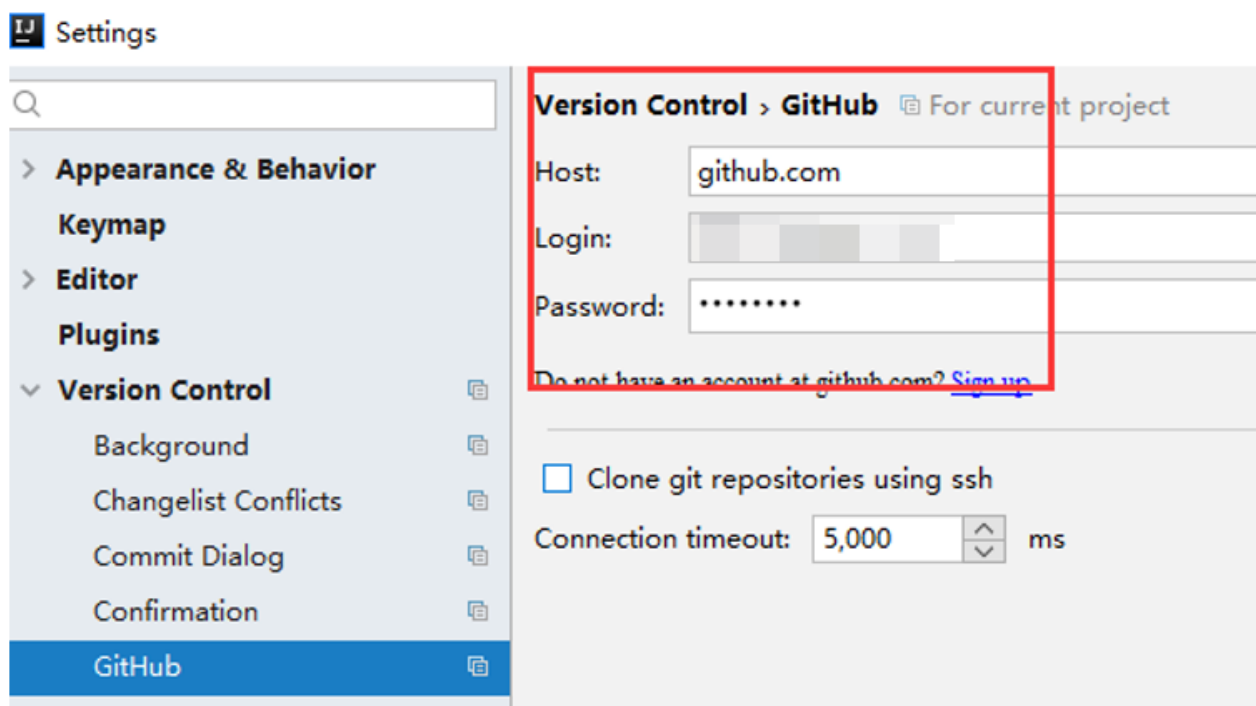
选择本地的Git.exe的位置



②

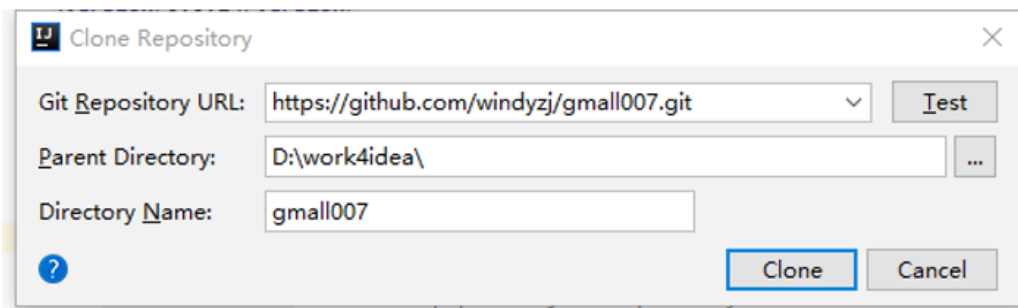
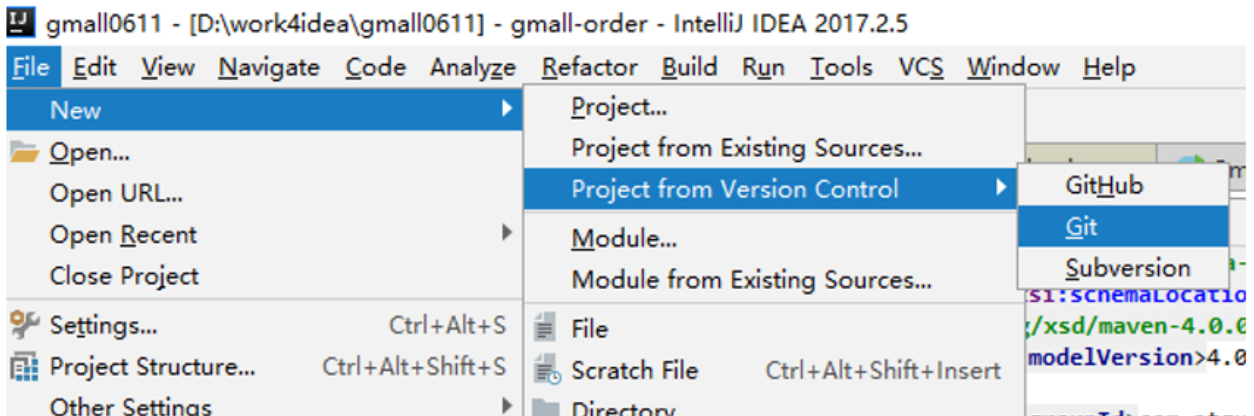
在idea中，settings → Version Control → GitHub

填写网站登录的用户名密码



③

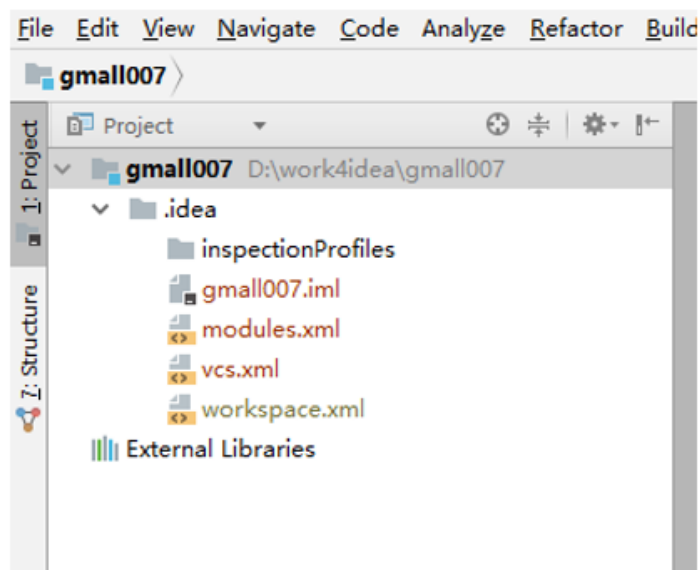
在idea中，通过版本控制软件创建项目



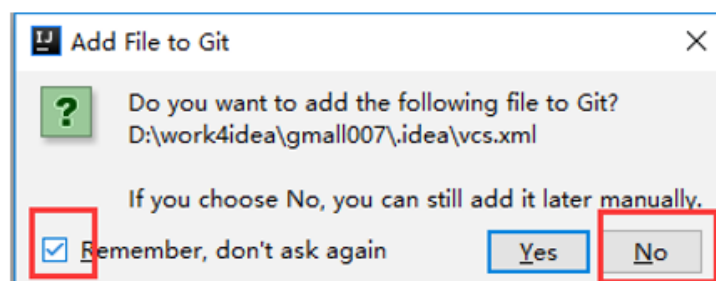
④

1) 从git上下载下来后，然后可能要新建项目，看要求来，但如果弹出让自动提交文件的提示，一定要选择NO，并在v中不再提示

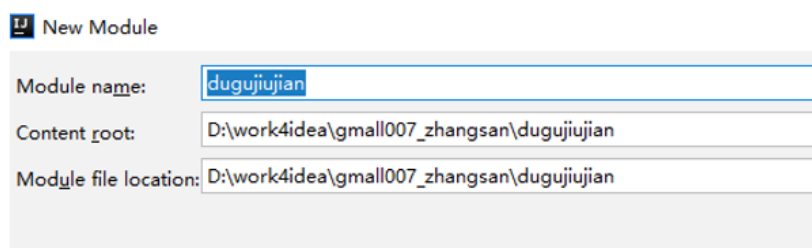
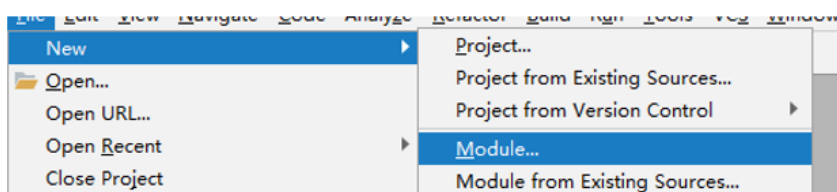
选择新建项目



有时候会弹出让你提交文件的提示，请选择No，并勾中不再提示。



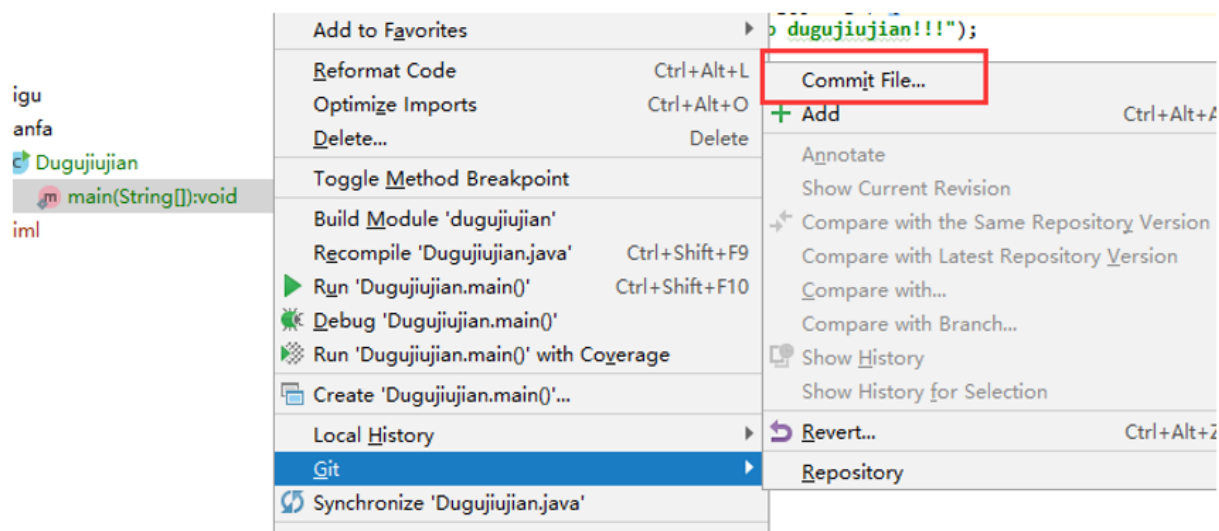
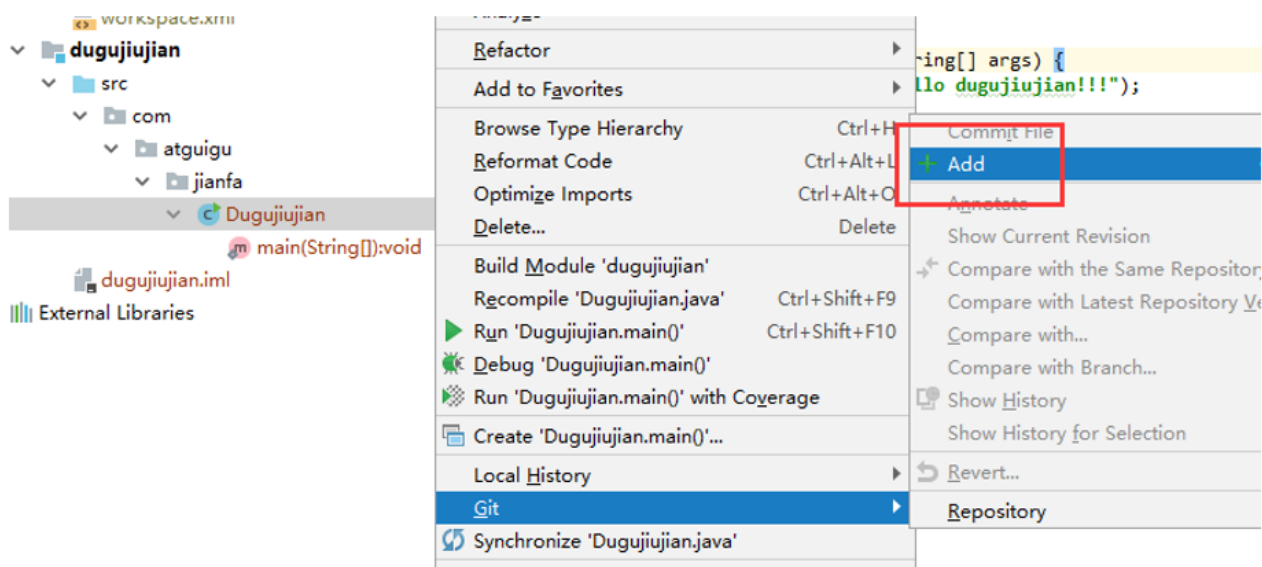
2) 还有一种：把clone下来的项目中的文件转换为模块

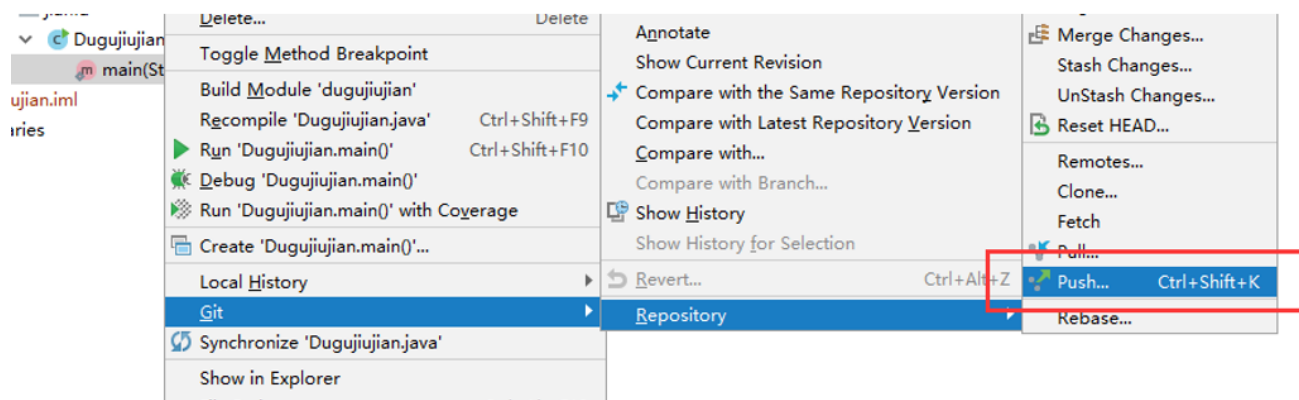
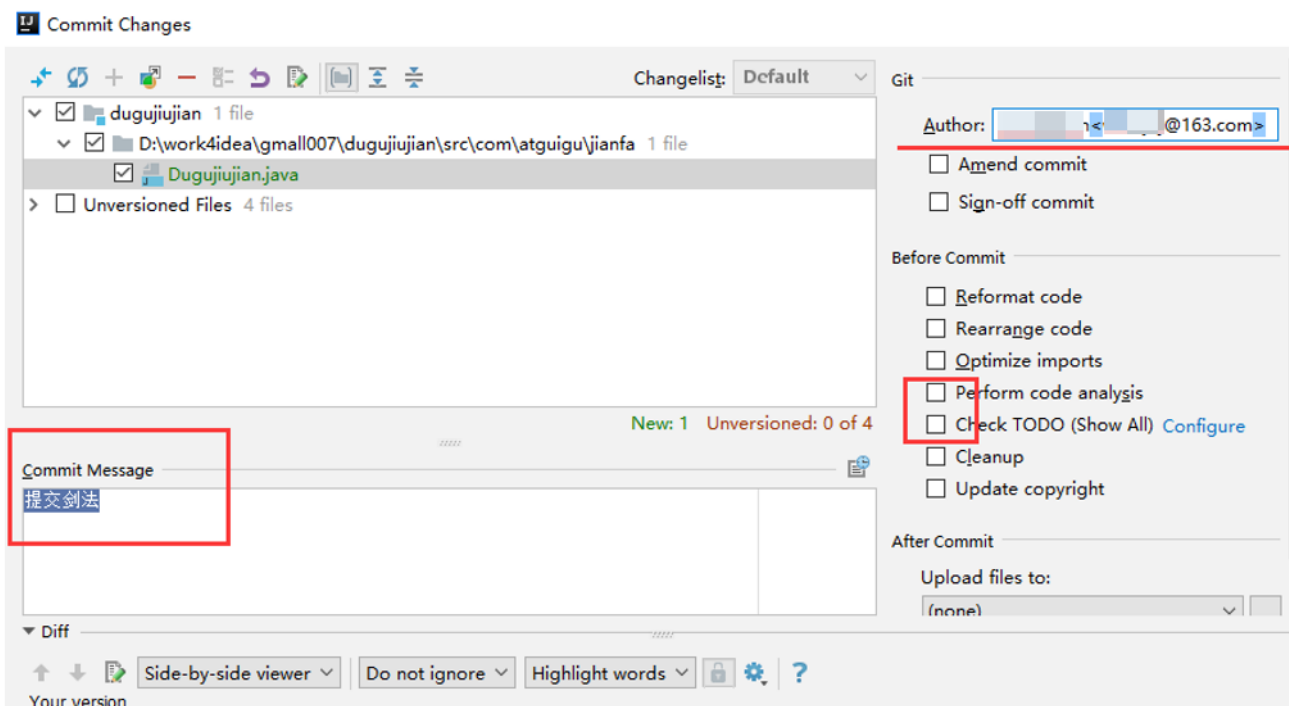


⑤

如果想往git上提交的话，可以如下操作：

- ① 选中想提交的（**务必不能选择所有的**）右键选中git找到add，添加到暂存区
- ② commit File，提交到本地库（这里要填写提交信息，加上作者，并去掉右边两项的提交之前代码分析等）
- ③ 推送到github上

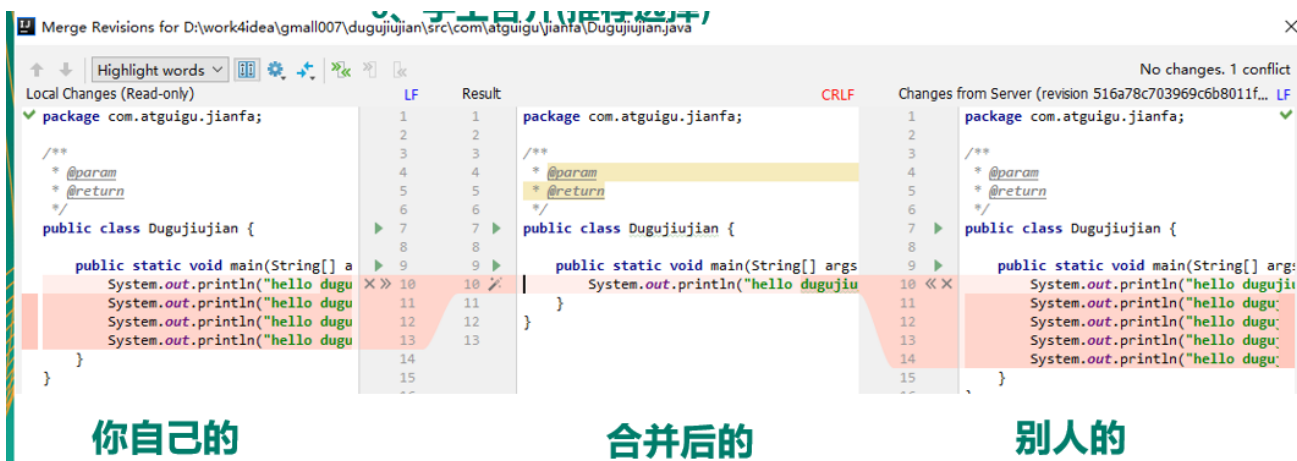




8、解决冲突

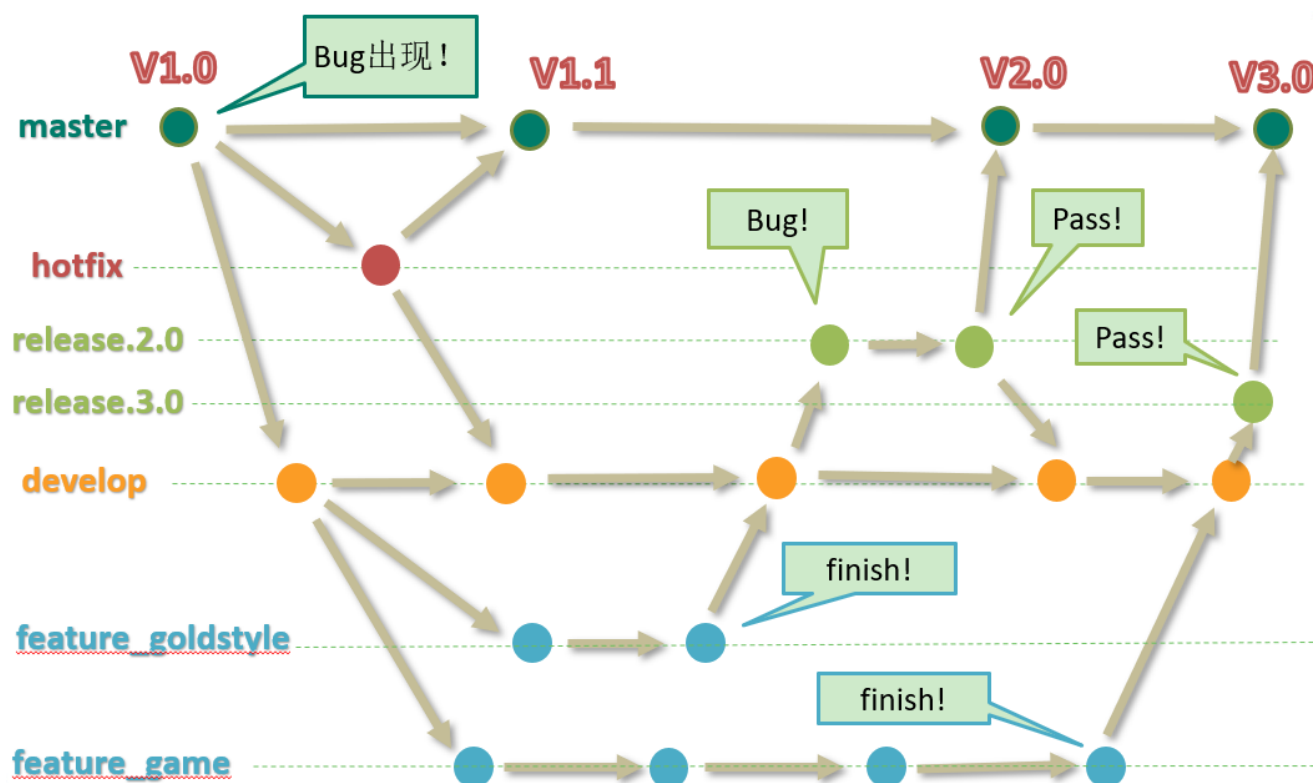
- 两个开发人员同样一块代码做了不同的修改，其中一个人提交了，另一个再提交就会报错
- 这个时候可以直接进行merge（idea会把最新的代码下载下来，然后和本地代码发生冲突）
- 这个时候会有3个选项：

- ① 直接以你本地代码为准覆盖掉别人的
- ② 直接以别人代码为准覆盖掉你的
- ③ 手工合并（一般大家都会选择这个）



- 编辑合并后，再次commit file
- 再次push

9、git工作流



•主干分支 master

主要负责管理正在运行的生产环境代码。永远保持与正在运行的生产环境完全一致。

•开发分支 develop

主要负责管理正在开发过程中的代码。一般情况下应该是最新的代码

•bug修理分支 hotfix

主要负责管理生产环境下出现的紧急修复的代码。从主干分支分出，修理完毕并测试上线后，并回主干分支。并回后，视情况可以删除该分支。

•发布版本分支 release

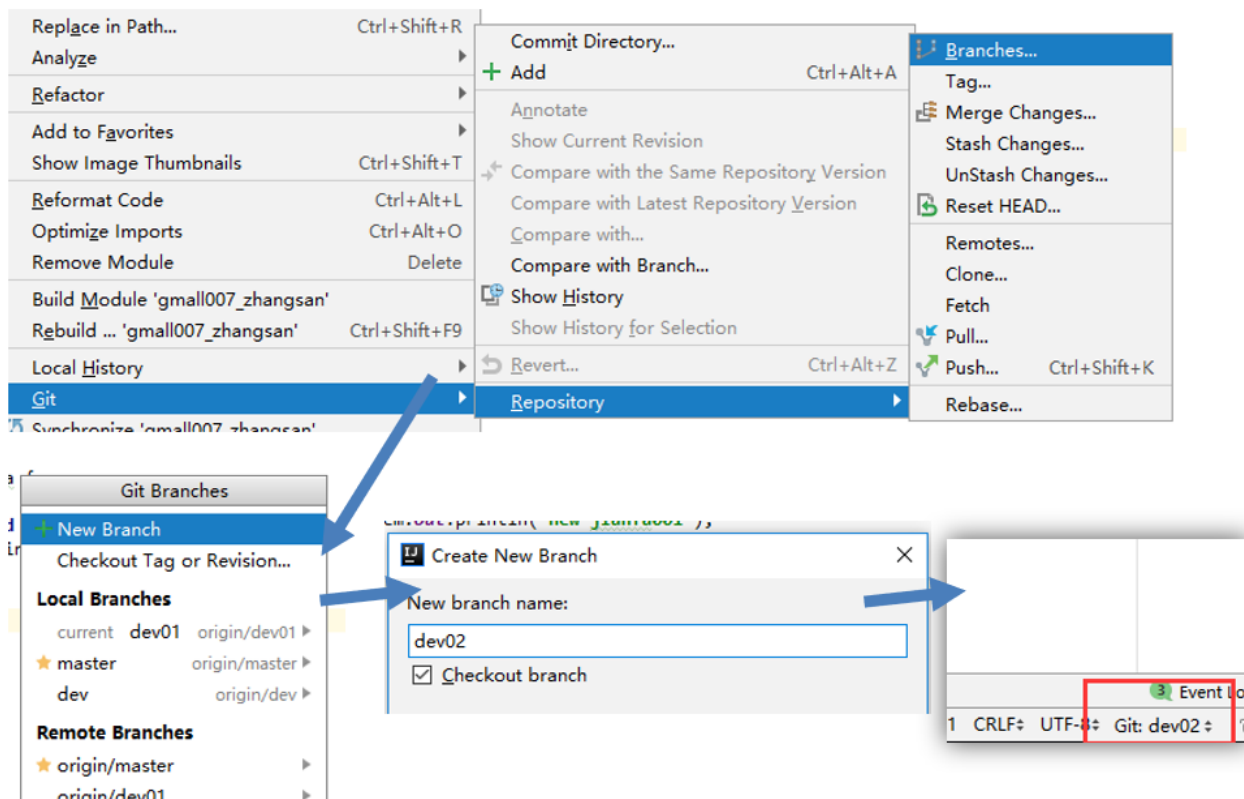
较大的版本上线前，会从开发分支中分出 **发布版本分支**，进行最后阶段的集成测试。该版本上线后，会 **合并到主干分支**。生产环境运行一段阶段较稳定后可以视情况删除。

•功能分支 feature

为了不影响较短周期的开发工作，一般把中长期开发模块，会从开发分支中 **独立** 出来。开发完成后会 **合并到开发分支**。

10、idea分支操作

① 创建分支

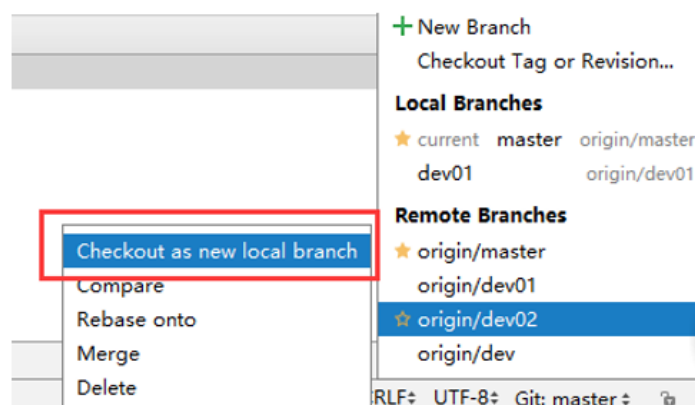


② 在分支上增加代码，并提交

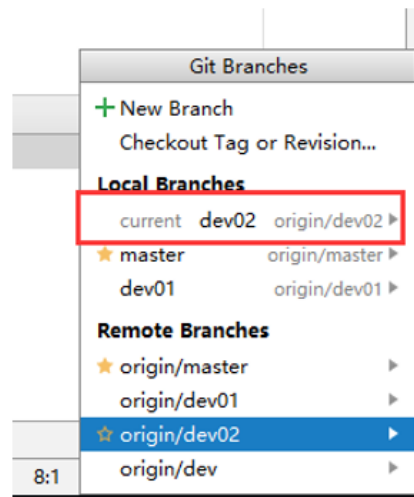
- ① 在这个分支下增加新代码，
- ② 本地提交，然后push到远端
- ③ 之后可以在github上看到新的分支被上传

③ 查看该分支内容

➤ 想要查看该分支内容

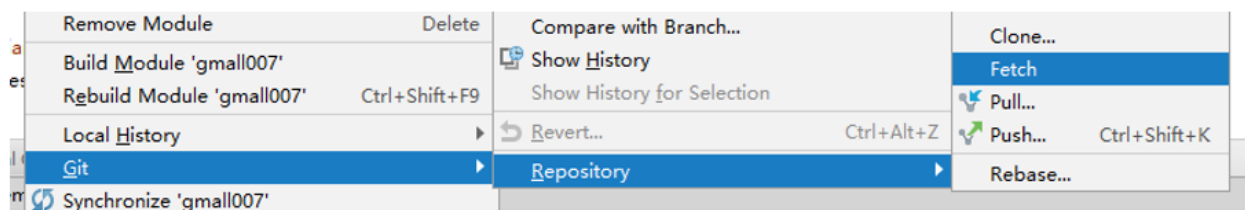


➤ 把远程分支下载下来变成本地分支

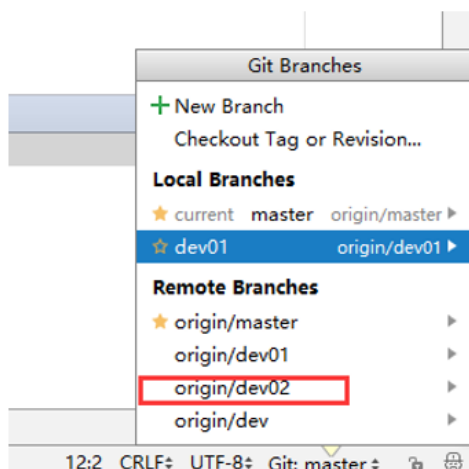


④ 其他程序员想看分支内容

➤ 另一个开发人员，想获得分支，把分支fetch下来



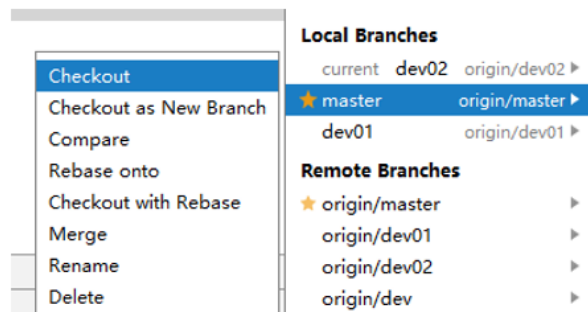
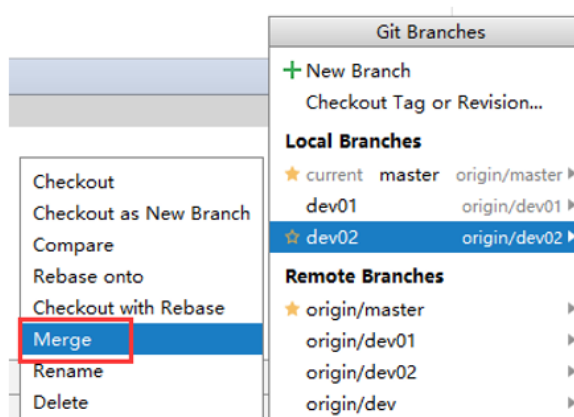
➤ 能看到远程分支



⑤ 无问题时候往主干合并

➤ 如果该分支内容修改没有问题，可以合并到主干

➤ 首先切换到主干master



在主干master分支上选择
要合并的分支进行Merge

最后把经过合并的master，push到远端即完成一个功能在主干上的提交（push之前需要先添加，再提交，再push）