Idea中Git的使用和两种类型的冲突解决

2017年06月16日 17:35:03 DayThinking 阅读数：32570

一、Git冲突解决

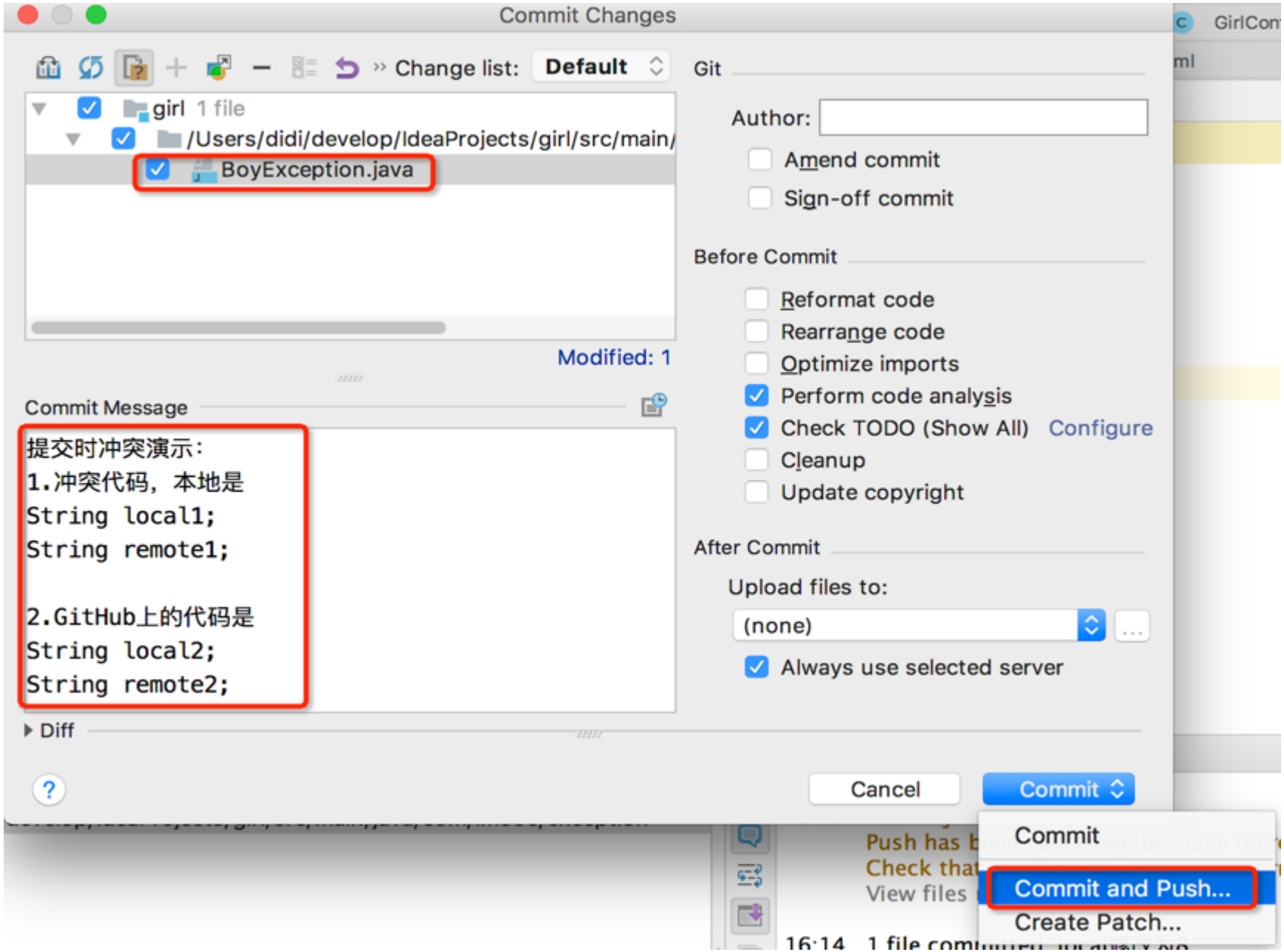
在idea开发工具中使用Git时，主要用到的快捷按钮如下五个：

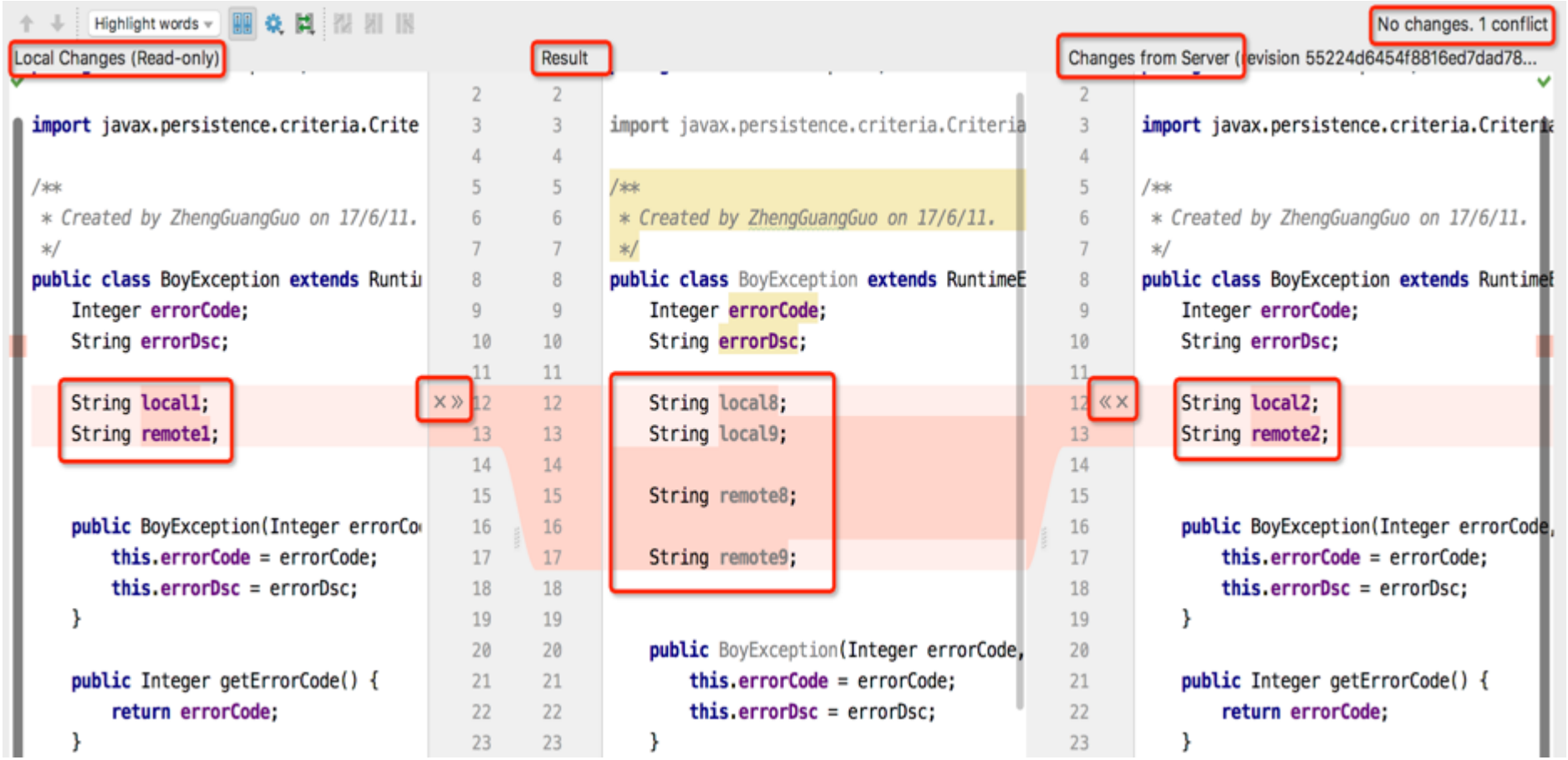
这五个按钮的使用说明及在idea中如何配置和使用git可参考https://github.com/DayThink/IntelliJ-IDEA-Tutorial/blob/newMaster/vcs-introduce.md

本文主要讲解在Idea中利用git遇到的两种冲突(提交代码时发生冲突和更新代码时发生冲突)以及解决方法，无论是那种冲突，只要发生冲突了，idea都会弹出一个图形化的merge对话框，在merge对话框上引导用户去解决冲突。

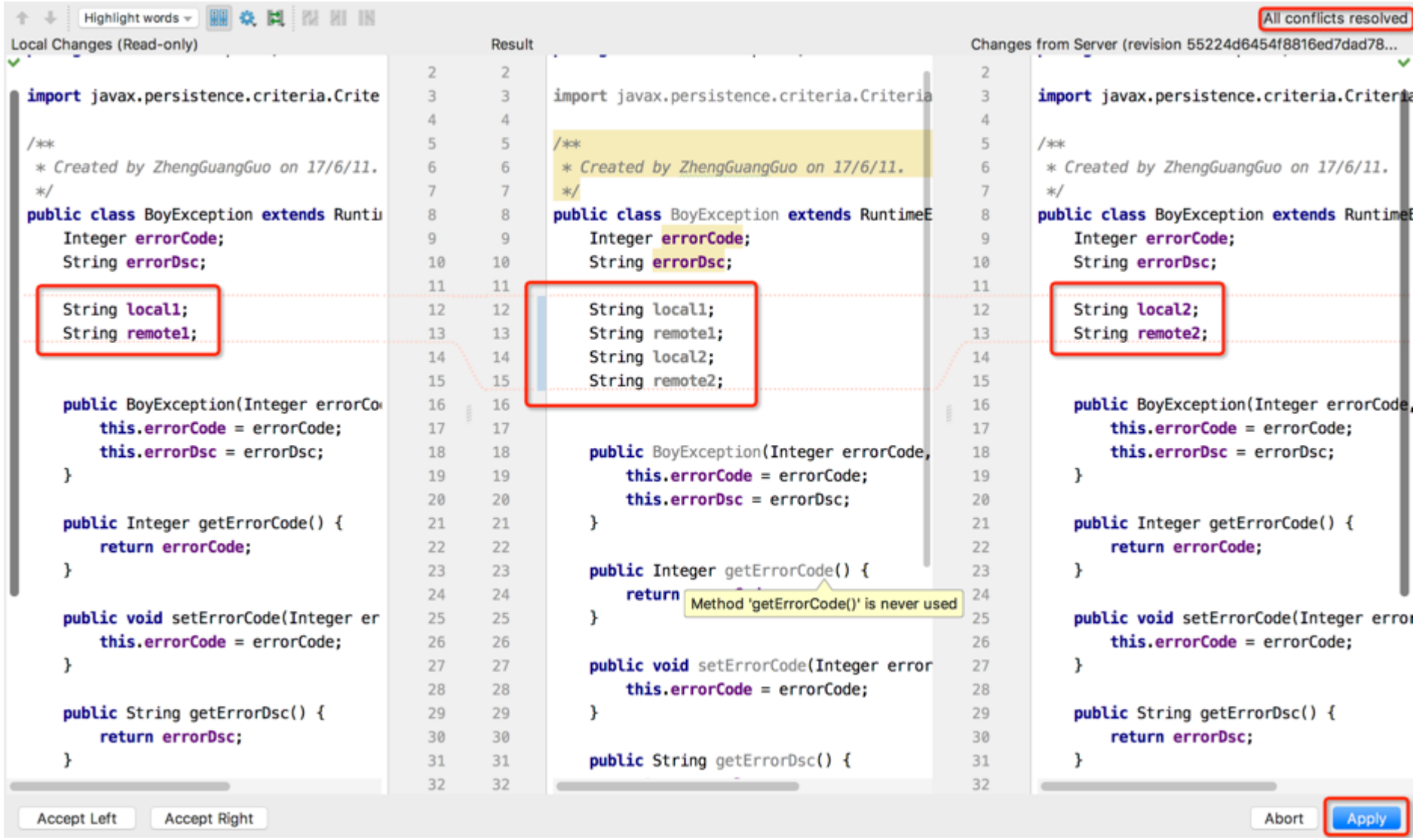
1.提交commit冲突：

如上图所示，当提交代码时发生冲突，此时操作步骤如下

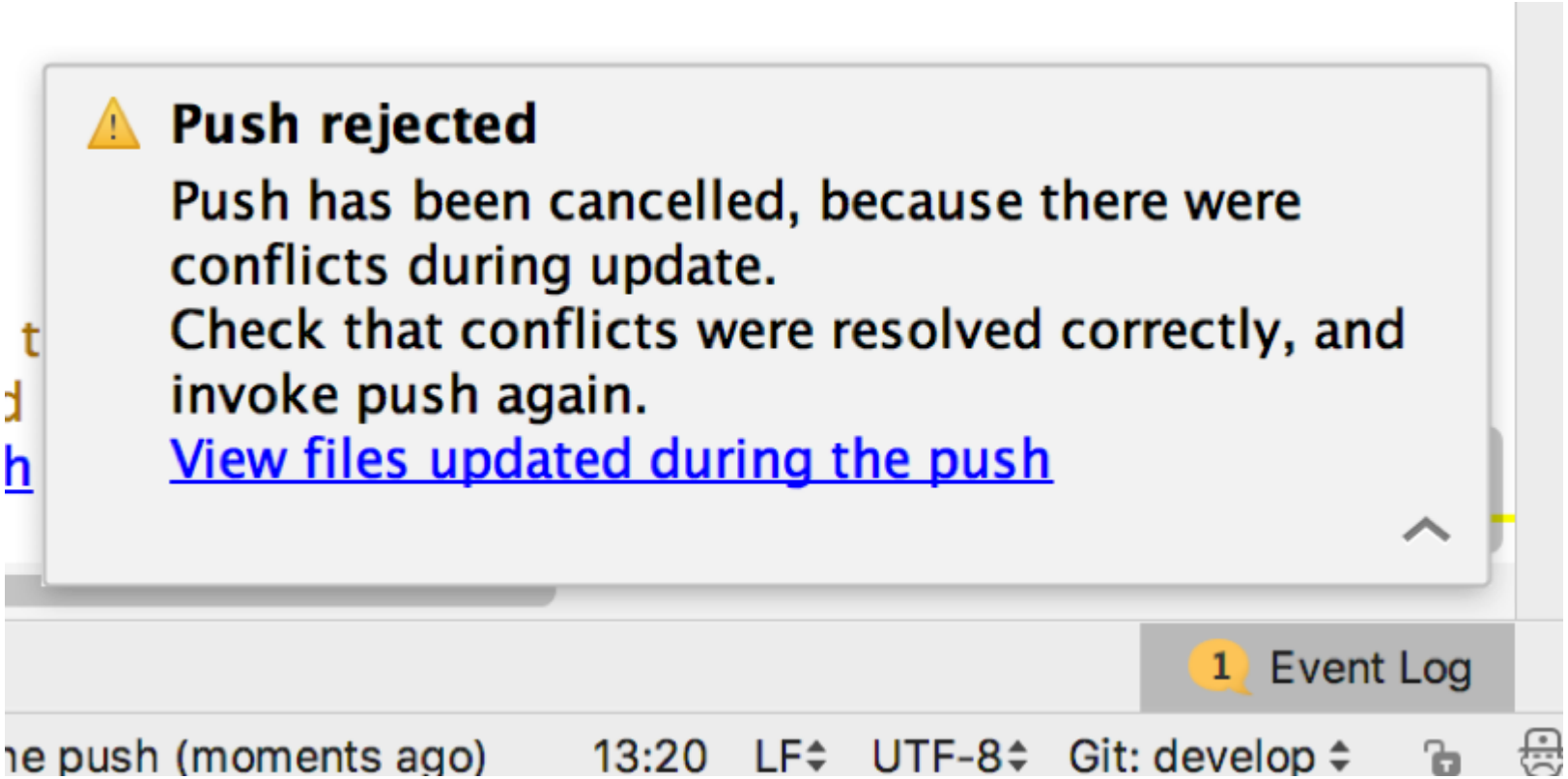

Commit Changes
Change list: Default
girl 1 file
/Users/didi/develop/IdeaProjects/girl/src/main/
BoyException.java
Modified: 1
Commit Message
提交时冲突演示:
1.冲突代码，本地是
String local1;
String remote1;
2.GitHub上的代码是
String local2;
String remote2;
Diff
Git
Author:
Amend commit
Sign-off commit
Before Commit
Reformat code
Rearrange code
Optimize imports
Perform code analysis
Check TODO (Show All)
Configure
Cleanup
Update copyright
After Commit
Upload files to:
(none)
Always use selected server
Cancel
Commit
Commit
Commit and Push...
Create Patch...



在这里我们希望将本地冲突代码和服务端的代码都保留，此时在上图中点击中间的两个红色按钮后即可将本地和服务端的冲突代码全部保留，在中间一栏result中科院预览到冲突解决后的代码情况，冲突处理完成后如下图

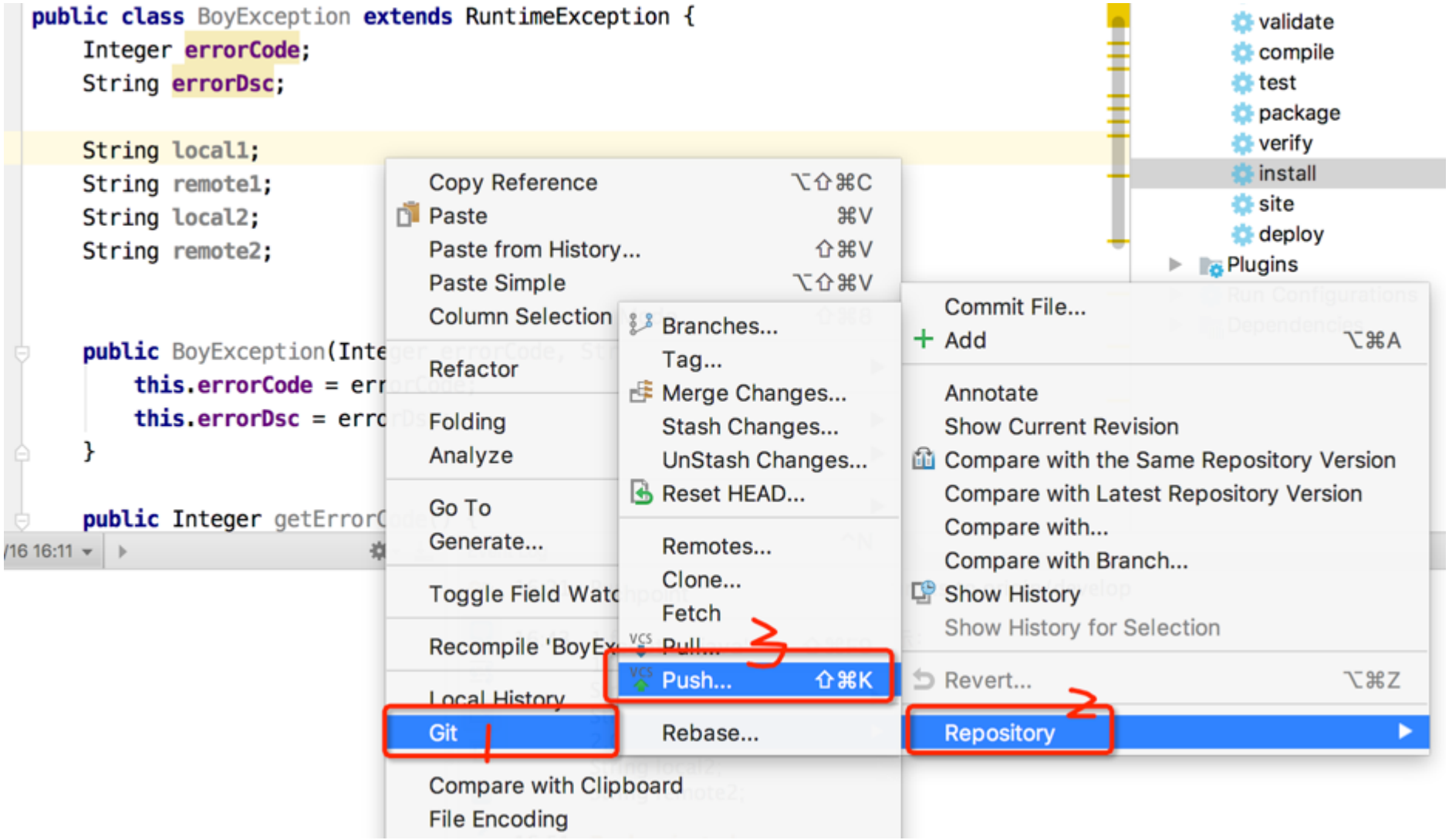


在上图merge对话框中点击右下角的apply时，一般会在idea的右下角出现如下的Push rejected提示对话框



上图对话框提示我们，在push的时候发生了失败，此时要注意，将本地代码更新提交到服务端，需要经历两个步骤：commit和push，commit相当于提交到了本地库，而push则是将本地库提交更新到服务端。根据上图提示push rejected，我们可以发现冲突已经在本地库commit时解决了，我们只需push一次即可，因此后续操作只需要push一次即可，如下：

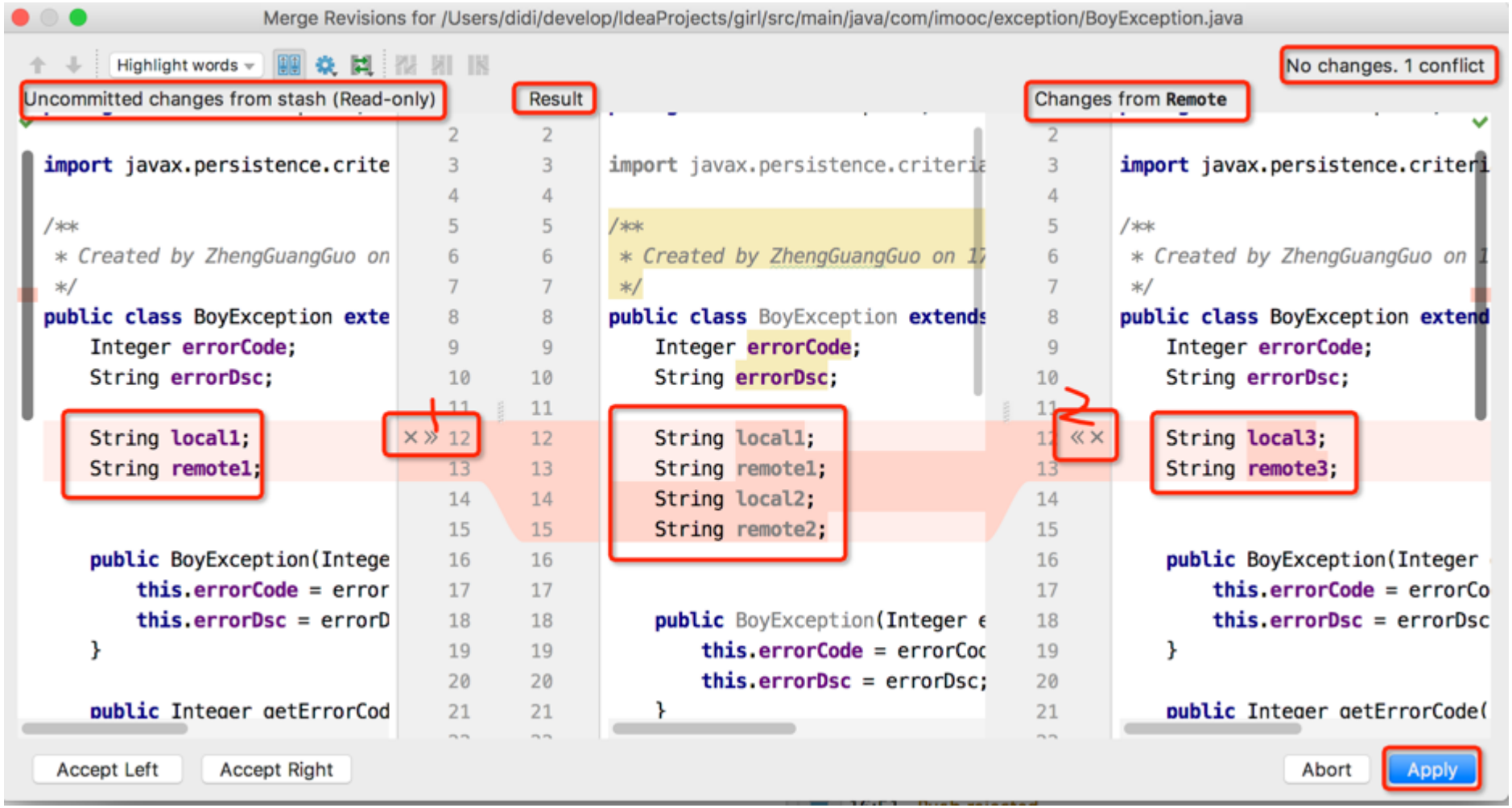
在冲突的文件上，右键



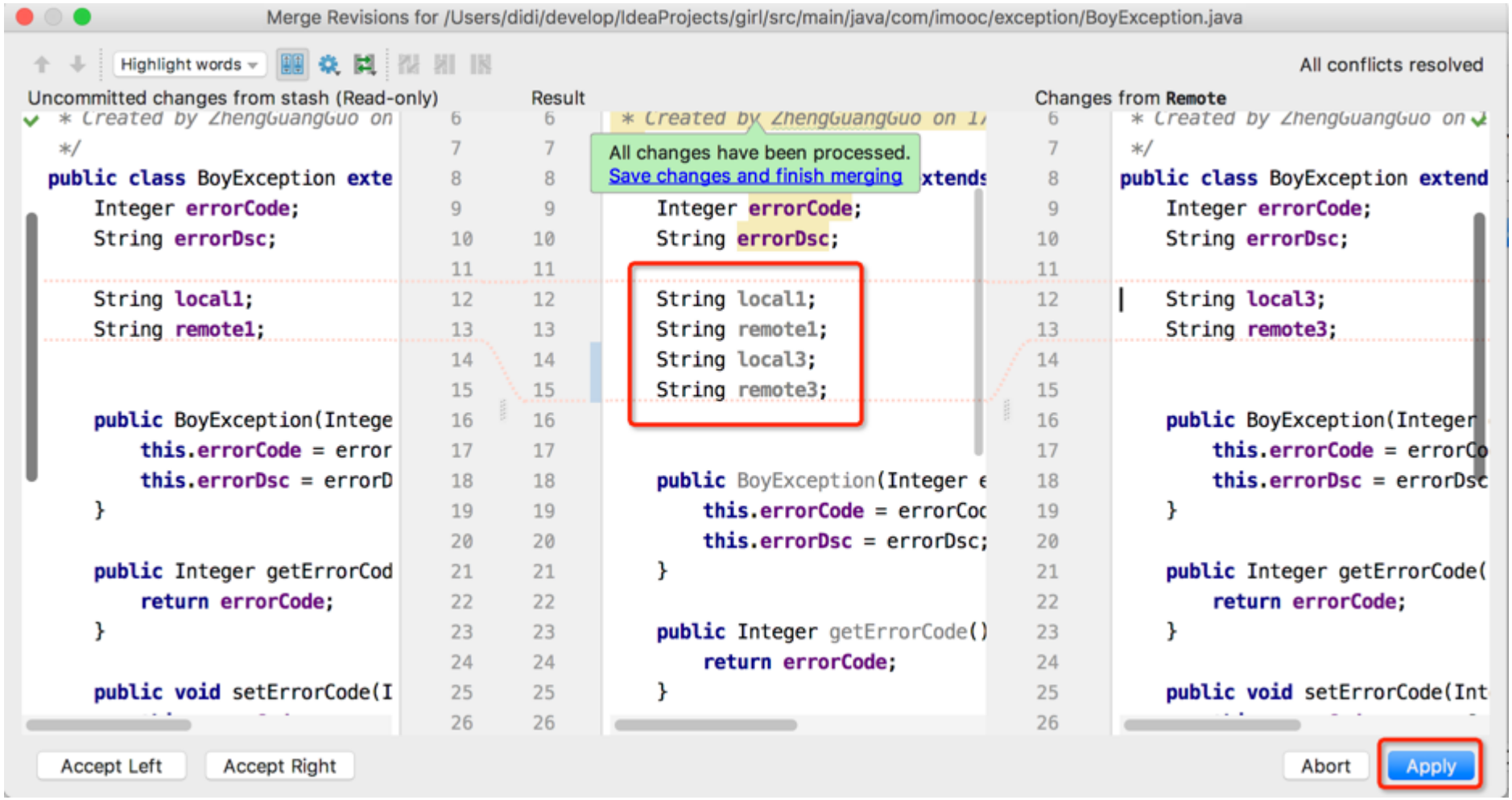
push完成后即可将本地库的文件提交到服务端。

2.更新update冲突

如上图所示，点击更新按钮时发现冲突，提示用户如下所示的merge页面上进行冲突的处理，比如此处我们希望将本地修改和远端服务器的修改均保留，因此按照如下图所示的红色1和2操作，合并到中间的result一栏中。



冲突解决后，如下所示：



至此，更新冲突就已解决完成。

二.Git命令的全家福手册

Git 常用命令速查表

```
master :默认开发分支        Head  :默认开发分支
origin :默认远程版本库      Head^ :Head 的父提交
```

创建版本库

```
$ git clone <url>              #克隆远程版本库
$ git init                     #初始化本地版本库
```

修改和提交

```
$ git status                   #查看状态
$ git diff                     #查看变更内容
$ git add .                    #跟踪所有改动过的文件
$ git add <file>               #跟踪指定的文件
$ git mv <old> <new>           #文件改名
$ git rm <file>                #删除文件
$ git rm --cached <file>       #停止跟踪文件但不删除
$ git commit -m “commit message”
                               #提交所有更新过的文件
$ git commit --amend           #修改最后一次提交
```

查看提交历史

```
$ git log                      #查看提交历史
$ git log -p <file>            #查看指定文件的提交历史
$ git blame <file>             #以列表方式查看指定文件
                               的提交历史
```

撤消

```
$ git reset --hard HEAD        #撤消工作目录中所有未提交
                               文件的修改内容
$ git checkout HEAD <file>     #撤消指定的未提交文件的修
                               改内容
$ git revert <commit>          #撤消指定的提交
```

分支与标签

```
$ git branch                   #显示所有本地分支
$ git checkout <branch/tag>    #切换到指定分支或标签
$ git branch <new-branch>      #创建新分支
$ git branch -d <branch>       #删除本地分支
$ git tag                      #列出所有本地标签
$ git tag <tagname>            #基于最新提交创建标签
$ git tag -d <tagname>         #删除标签
```

合并与衍合

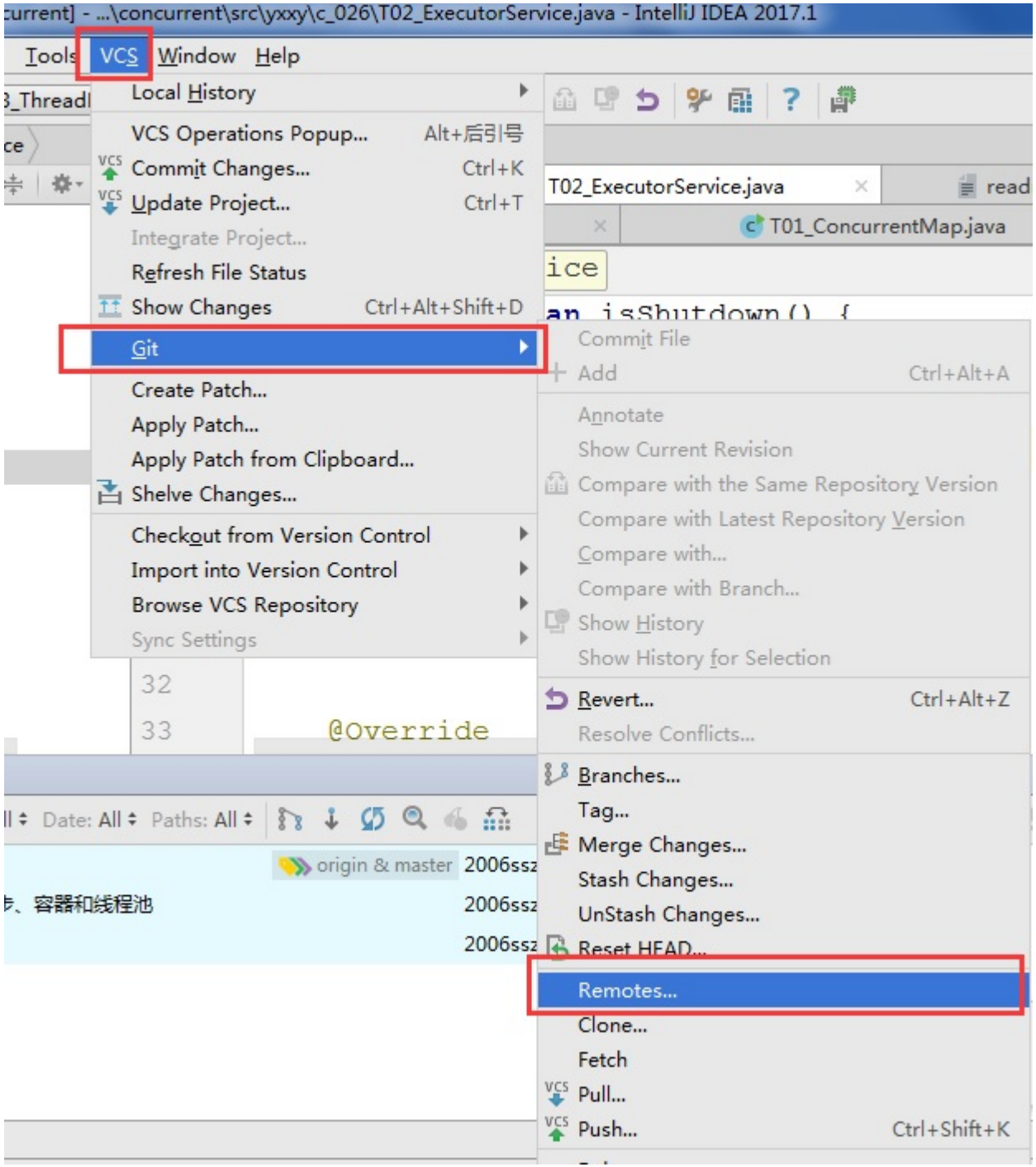
```
$ git merge <branch>           #合并指定分支到当前分支
$ git rebase <branch>          #衍合指定分支到当前分支
```

远程操作

```
$ git remote -v                #查看远程版本库信息
$ git remote show <remote>     #查看指定远程版本库信息
$ git remote add <remote> <url>
                               #添加远程版本库
$ git fetch <remote>           #从远程库获取代码
$ git pull <remote> <branch>   #下载代码及快速合并
$ git push <remote> <branch>   #上传代码及快速合并
$ git push <remote> :<branch/tag-name>
                               #删除远程分支或标签
$ git push --tags              #上传所有标签
```

三、idea中查看远程Git仓库的地址

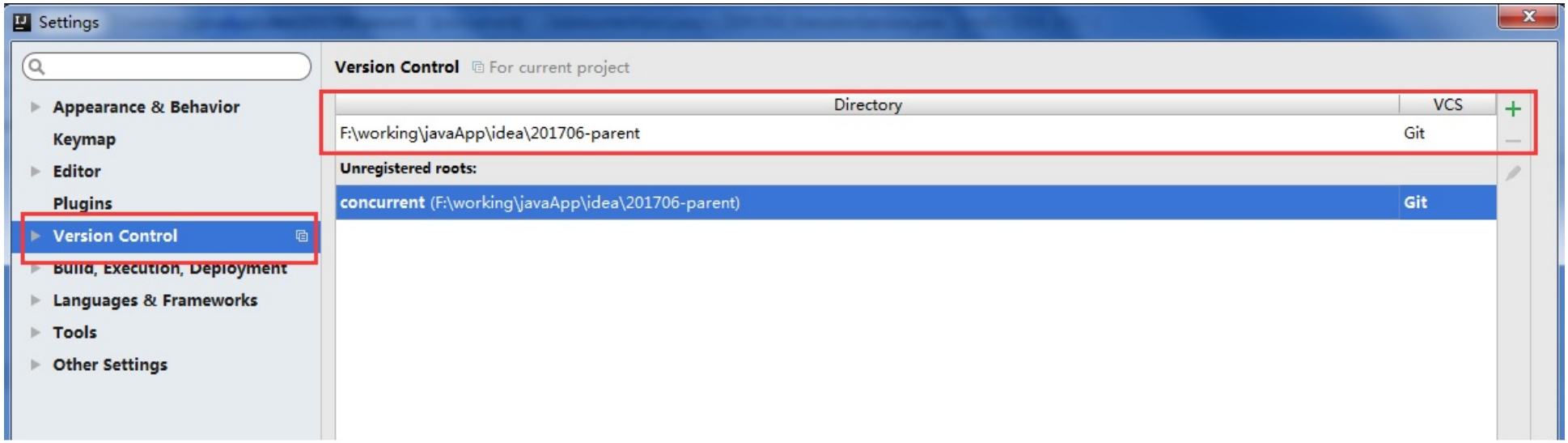
有时我们需要将本地的代码与远程Git仓库脱离绑定关系或重新绑定到另一个Git仓库上，此时在idea中查看或删除本地代码对应Git仓库的方法：


current] - ...\concurrent\src\yxxy\c_026\T02_ExecutorService.java - IntelliJ IDEA 2017.1
Tools
VCS
Window
Help
Local History
VCS Operations Popup... Alt+后引号
Commit Changes... Ctrl+K
Update Project... Ctrl+T
Integrate Project...
Refresh File Status
Show Changes Ctrl+Alt+Shift+D
Git
Create Patch...
Apply Patch...
Apply Patch from Clipboard...
Shelve Changes...
Checkout from Version Control
Import into Version Control
Browse VCS Repository
Sync Settings
T02_ExecutorService.java
T01_ConcurrentMap.java
ice
an isShutdown() {
32
33
@Override
Commit File
Add Ctrl+Alt+A
Annotate
Show Current Revision
Compare with the Same Repository Version
Compare with Latest Repository Version
Compare with...
Compare with Branch...
Show History
Show History for Selection
Revert... Ctrl+Alt+Z
Resolve Conflicts...
Branches...
Tag...
Merge Changes...
Stash Changes...
UnStash Changes...
Reset HEAD...
Remotes...
Clone...
Fetch
Pull...
Push... Ctrl+Shift+K
Date: All
Paths: All
origin & master
2006ssz
2006ssz
2006ssz
、容器和线程池

如果本地的仓库绑定了两个远端的Git仓库，那么在弹出的Remotes对话框中会有两条远端Git仓库的记录。

四.idea中查看本地Git仓库的地址

查看与远程Git仓库绑定的本地目录，有时候本地目录和子目录都会绑定到一个远端的Git仓库。



五.用Git命令行(不在idea操作)上传本地仓库到Git远程仓库

如果是新项目，创建 git 仓库:

```
mkdir SpringBootInAction
cd SpringBootInAction
git init
##将当前目录下的所有文件添加到git仓库
git add .
git commit -m "first commit"
##首先要建好一个远端仓库(URL),如https://xxx/SpringBootInAction.git
```

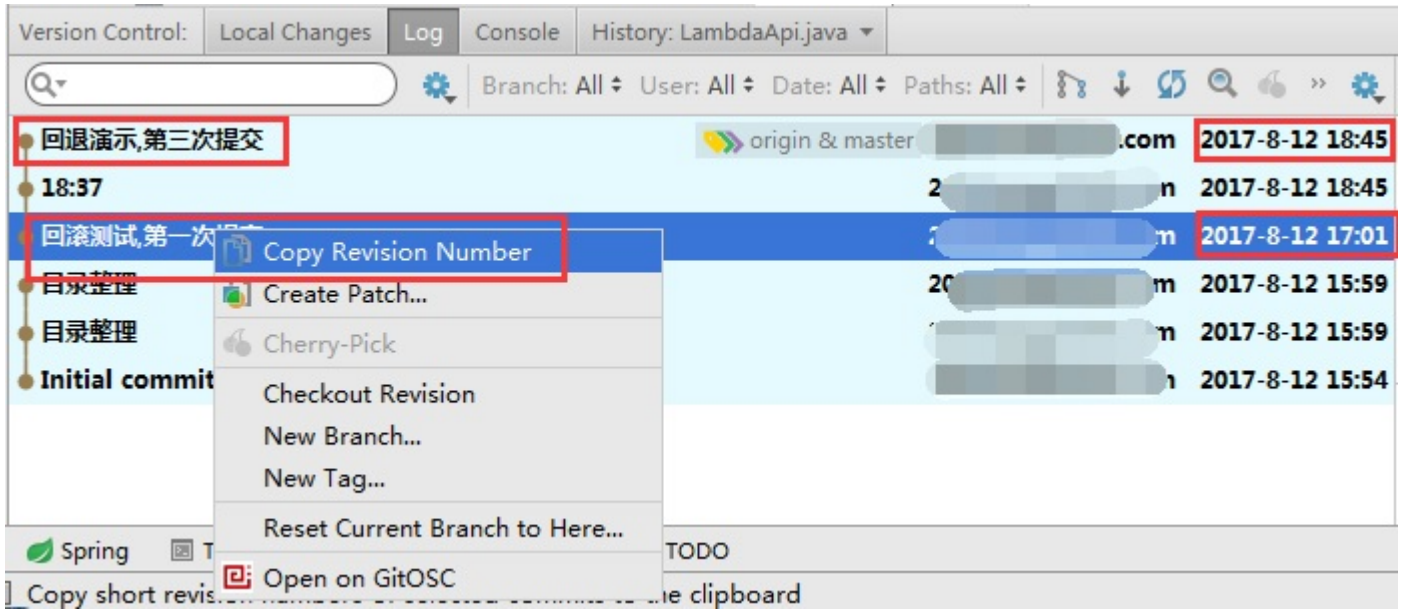
```
git remote add origin https://xxx/SpringBootInAction.git | git push -u -f origin master
```

如果是已有项目

```
cd existing_git_repo
git remote add origin https://xxx/SpringBootInAction.git
git push -u origin maste
```

六.Git版本回退

首先，根据git提交的历史记录中确定回退的版本



将上述拷贝的Revision Number记录下来，如：4b91ea12de20cdf97eccb6bc9125bbc2fef79b17
备注：后面两步在idea中好似不能图形化操作界面，只能用命令行了

其次，恢复到指定的commit hash
git reset --hard resetVersionHash
eg:git reset --hard 4b91ea12de20cdf97eccb6bc9125bbc2fef79b17

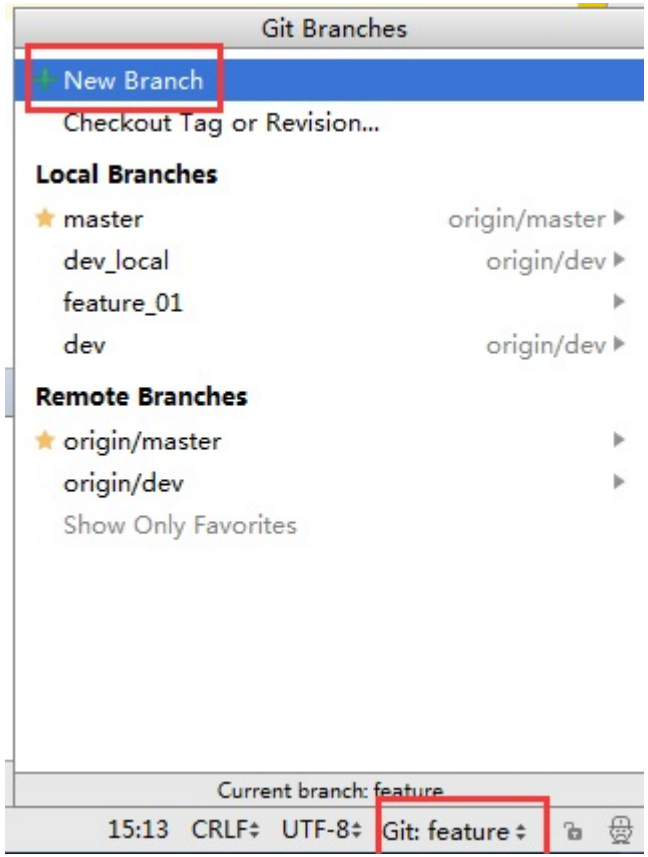
最后，把当前分支强制提交到远程
git push -f origin currentBranch
eg: push -f origin master

以上三步操作后，就将本地和git远端的代码从2017-8-12 18:45回退到了2017-8-12 17:01。

七、分支操作

7.1.新建分支

1.首先在本地新建一个分支，如dev;



2.执行git push origin dev:dev;或git push origin dev;

上面命令表示，将本地的dev分支推送到origin主机的dev分支，如果后者不存在，则会被新建。

git push命令用于将本地分支的更新，推送到远程主机，它的格式如下：

$ git push <远程主机名> <本地分支名>:<远程分支名>

如果省略远程分支名，则表示将本地分支推送与之存在”追踪关系”的远程分支(通常两者同名)，如果该远程分支不存在，则会被新建。

$ git push origin master

上面命令表示，将本地的master分支推送到origin主机的master分支，如果后者不存在，则会被新建。

如果省略本地分支名，则表示删除指定的远程分支，因为这等同于推送一个空的本地分支到远程分支。

$ git push origin :master

等同于

$ git push origin --delete master

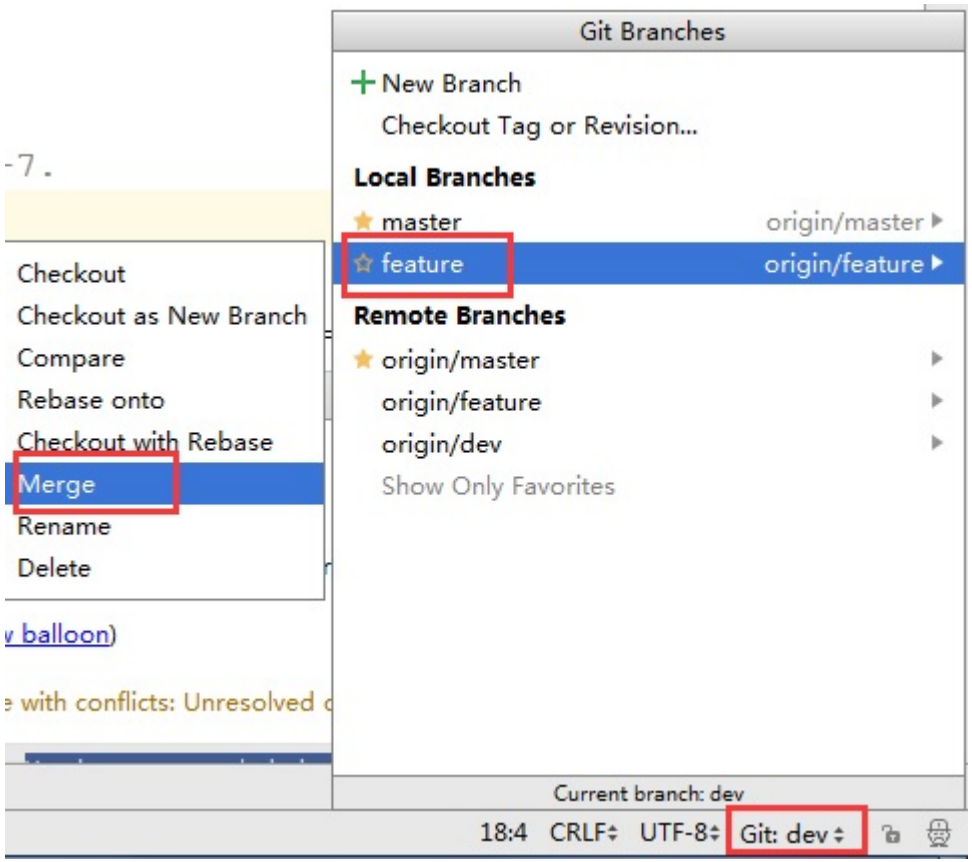
上面命令表示删除origin主机的master分支。
如果当前分支与远程分支之间存在追踪关系，则本地分支和远程分支都可以省略。
$ git push origin
上面命令表示，将当前分支推送到origin主机的对应分支。
如果当前分支只有一个追踪分支，那么主机名都可以省略。
$ git push
最后需要注意与git pull的区别，从远程分支名拉到(合并)本地分支名上：
$ git pull <远程主机名> <远程分支名>:<本地分支名>

7.2.本地与origin分支的对应设置

查看本地分支与远端origin的分支对应情况
$ git remote show origin设置本地分支与远端分支的对应关系，如:将本地dev分支与远端的dev分支进行绑定
git branch --set-upstream dev origin/dev

7.3分支合并

如下所示，将feature分支合并到dev分支，之后将合并后的dev分支push一次到origin即可。



7.4.远端origin分支查看及操作

```
git remote -v
git remote add origin_didi git@git.xiaojukeji.com:sec/ews-parent.git
git remote -v
git remote --help
git remote remove origin
git remote -v
git remote rename origin_didi origin
git remote -v
git push origin master
```

7.5.一个case

git merge [分支名] 合并指定分支到当前分支
分支切换利器，我们经常会遇到，正在一个分支上开发，写了很多东西，其中提交了一部分，一部分正在写，还没写完，这时突然来个线上问题，需要我们切换到线上分支去处理，我们怎么办，提交？还没写完，还不能通过编译，不提交，放弃?更不可能 辛苦工作一天了。
这时就需要下面这个命令，特别长用
git stash 暂时将未提交的变化暂存，形成类似一个快照，然后就可以随意切换到其他分支了。
git stash pop 当在其他分支解决完问题，切换到当前分支，执行该命令，之前的暂存快照又恢复了，可以继续工作了

参考文章

1.码云平台帮助文档
2.史上最简单的 IntelliJ IDEA 教程
3.史上最简单的 GitHub 教程
4.git动画演示（答案查看方式:在页面左侧的模拟控制台下输入show solution或直接参考这个网站 答案 ）
5.git
6.Git 常用命令速查手册