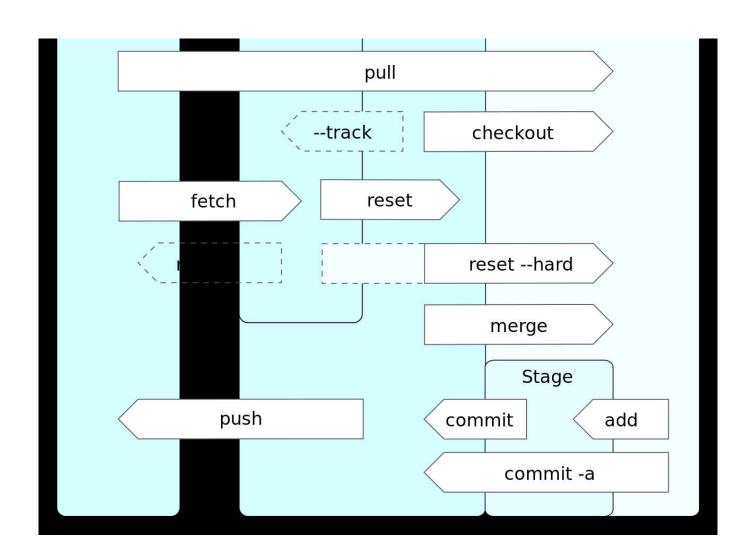
- *git 版本管理方法
- 在server上创建代码仓库
- 使用Pycharm 同步server 的代码
- Pycharm 中提交本地的git 信息
- 将本地版本提交到server上
- 建立分支
- 如何解决冲突?
 - Step1: 首先,选要将分支合并到哪个版本:点击pycharm右下角的按钮,切换版本
 - o Step2: 选择合并
 - Step3: 将合并的版本提交到server中即可即可
- 其他
 - 关于pycharm中git 的文件颜色
 - o Git发展史
 - cvs –svn
 - 集中式VS分布式的版本控制
 - o git 的运行方式
 - o 常用的git服务器

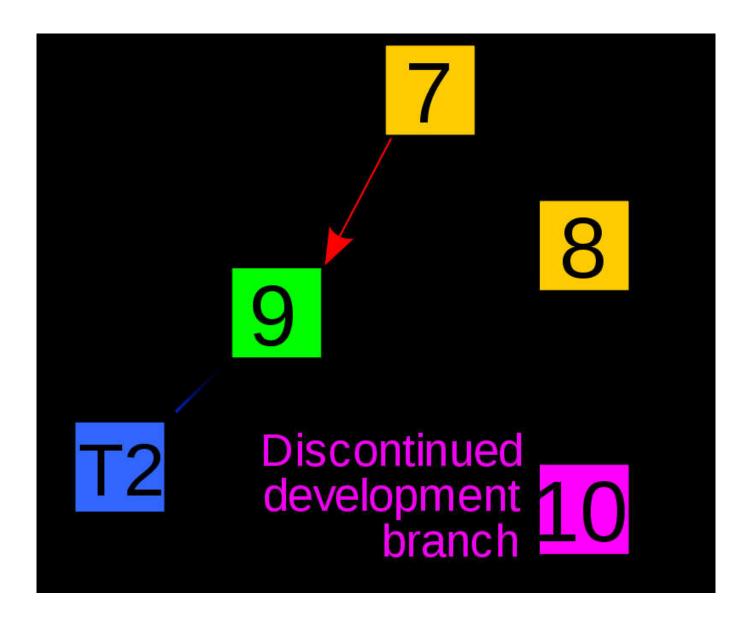
为了更好地管理目前公司内的源码版本,让大家更好的协同工作,利用业余时间学习了git的使用方法。

*git 版本管理方法

Remote Clone Branches Working Files



Trunks 2 Branches 3 Merges 5 Tags



在server上创建代码仓库

输入URL, 点击"版本库" --> "创建版本库", 流程如下图所示:

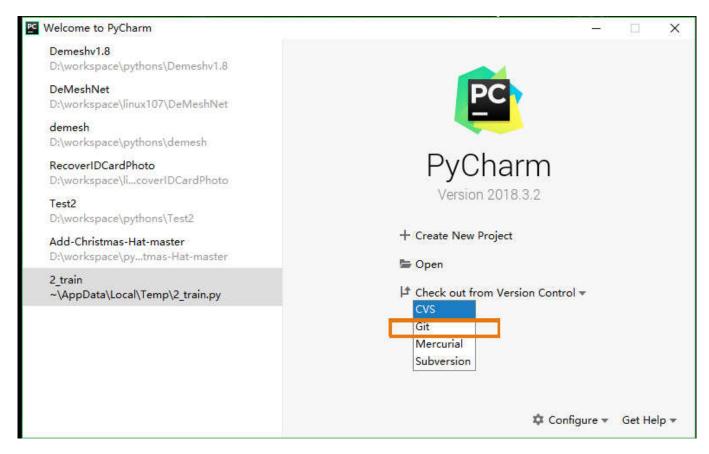


面跳转,显示项目信息: **这里需要注意版本库的地址URL,在后面的项目同步中会用到:ssh://lizhen@10.10.7.107:29418/~lizhen/Test.git **

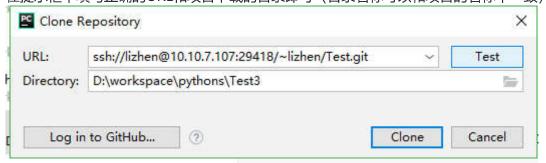


使用Pycharm 同步server 的代码

Pycharm提供了有关版本控制的客户端工具: git、svn、cvs等等;使用方法如下:在Pycharm的welcome界面点击 checkout from version control-->git

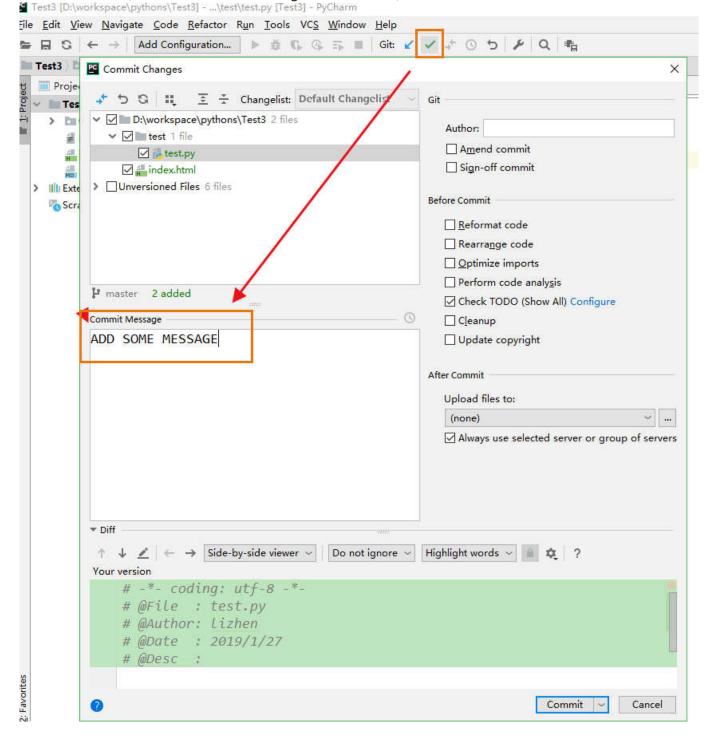


在提示框中填写正确的URL和项目下载的目录即可(目录名称可以和项目的名称不一致)



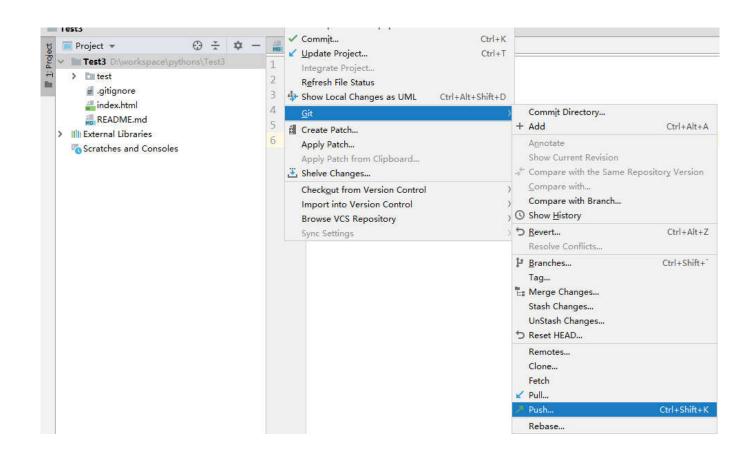
Pycharm 中提交本地的git 信息

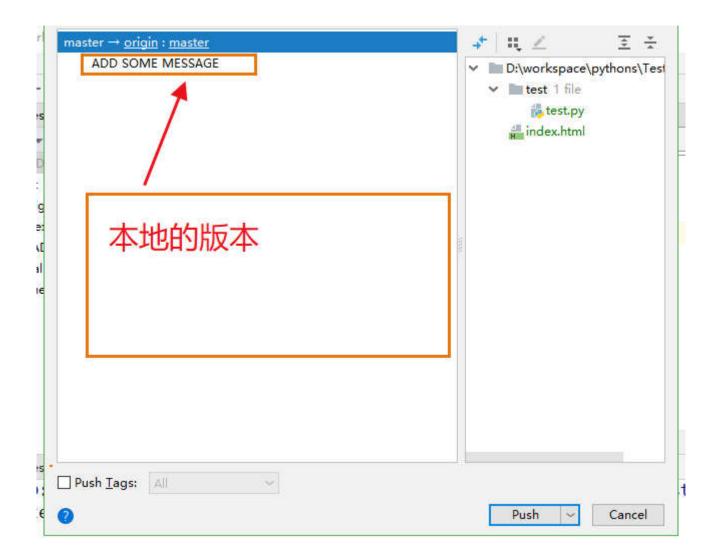
点击 view->选中"toolbar",此时在窗口的菜单栏下方出现了git工具栏,按照下图方式操作,即可完成本地的版本信息提交(注意:本地版本的message信息一定要写)。此时本地的版本与server上的不再同步了。



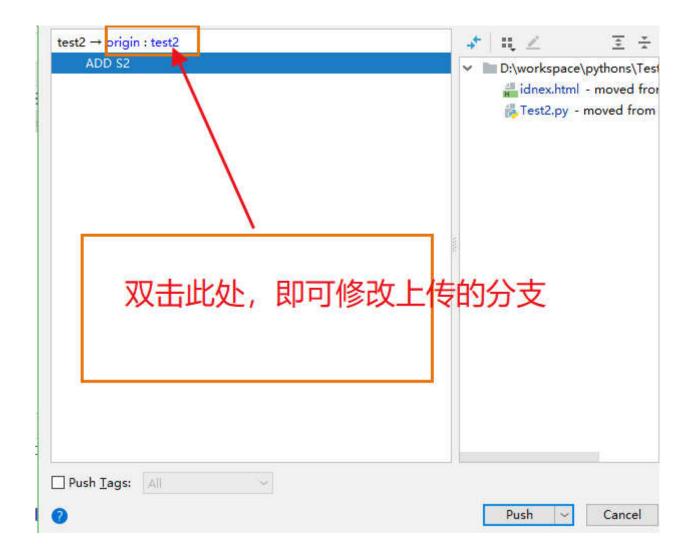
将本地版本提交到server上







建立分支



如何解决冲突?

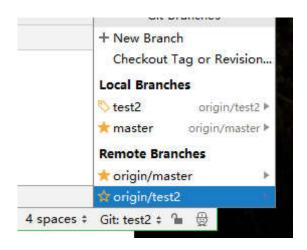
冲突出现的原因是 server上的内容和本地的内容完全不一致导致的。可以采用两种方式解决:

- 1. 建立分支
- 2. 将本地的版本合并到master 版本中(首先建立分支, 然后合并分支)

合并分支的过程如下所示:

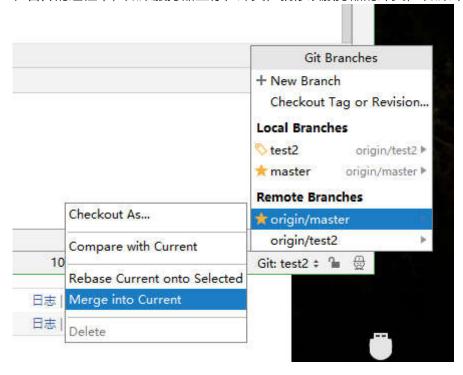
Step1: 首先,选要将分支合并到哪个版本:点击pycharm右下角的按钮,切换版本





Step2: 选择合并

在合并的过程中,如果服务器上存在冲突,就修改服务器的冲突;如果本地出现的冲突,就该本地冲突



Step3:将合并的版本提交到server中即可即可

其他

关于pycharm中git 的文件颜色



Git发展史

cvs -svn

在2002年以前,世界各地的志愿者把源代码文件通过diff的方式发给Linus,然后由Linus本人通过手工方式合并代码!你也许会想,为什么Linus不把Linux代码放到版本控制系系统呢?不是有CVS、SVN这些免费的版本控制系统吗?因为Linus坚定地反对CVS和SVN,这些集中式的版本控制系统不但速度慢,而且必须联网才能使用。有一些商用的版本控制系统,虽然比CVS、SVN好用,但那是付费的,和Linux的开源精神不符。

Linus花了两周时间用C写了一个分布式版本控制系统,这就是Git。一个月之内,Linux系统的源码已经由Git管理了。Git也迅速成为最流行的分布式版本控制系统,尤其是2008年,GitHub网站上线了,它为开源项目免费提供Git存储,无数开源项目开始迁移到GitHub,包括jQuery,PHP,Ruby等等。

集中式VS分布式的版本控制

1) 集中式版本控制系统,

版本库是集中存放在中央服务器的,你干活的时候,用的都是自己的电脑,所以要先从中央服务器取得最新的版本,然后开始干活,干完活了,再把自己的活推送给中央服务器。中央服务器就好比是一个图书馆,你要改一本书,必须先从图书馆借出来,然后回到家自己改,改完了,再放回图书馆。集中式版本控制系统最大的弊端就是必须联网才能工作,如果在局域网内还好,速度够快,可如果在互联网上,遇到网速慢的话,可能提交个10M的文件就需要5分钟,这还不得把人给憋死啊。

总之是一句话: 使用之前,需要先从server上同步代码,如果出现冲突,需要修改自己的代码,或者修改服务器的代码,解决了冲突以后才能提交---》**服务器和本地始终是只有一个版本的**

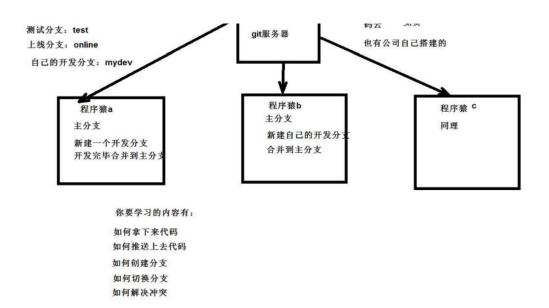
(2) 分布式版本控制系统

没有"中央服务器",每个人的电脑上都是一个完整的版本库,这样,你工作的时候,就不需要联网了,因为版本库就在你自己的电脑上。既然每个认电脑上都有个完整的版本库,那多个人如何协作呢?比如说你在自己电脑上改了文件A,你的同事也在他的电脑上改了文件A,这时,你们俩之间只需把各自的修改推送给对方,就可以互相看到对方的修改了。和集中式版本控制系统相比,分布式版本控制系统的安全性要好很多,因为每个电脑里都有完整的版本库,某个人的电脑坏掉了不要紧,而集中式版本控制系统的中央服务器要是出了问题,所有人都没法干活了

总之是一句话: 使用之前,需要先从server上同步代码,如果出现冲突,需要修改自己的代码,或者修改服务 器的代码,解决了冲突以后才能提交---**》服务器和本地的版本可以不同步,也可以提交与Sever上不同的版 本,即是从master上建立分支**

git 的运行方式

主分支: master 开发分支: development 中央服务器 coding 免费



https://blog.cedn.net/770HFLL020D

常用的git服务器

目前可以创建私人代码仓库的网站:

gitlab

github

git