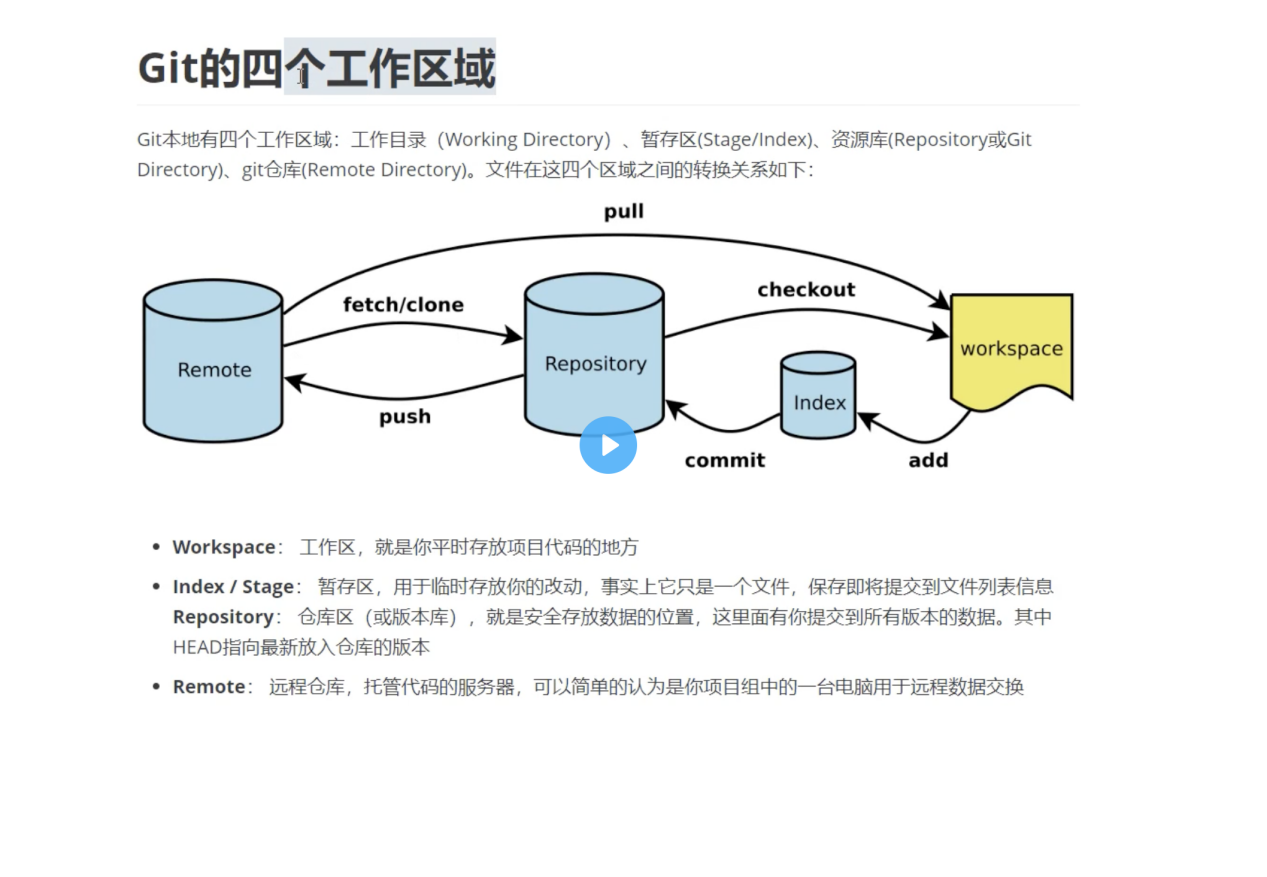
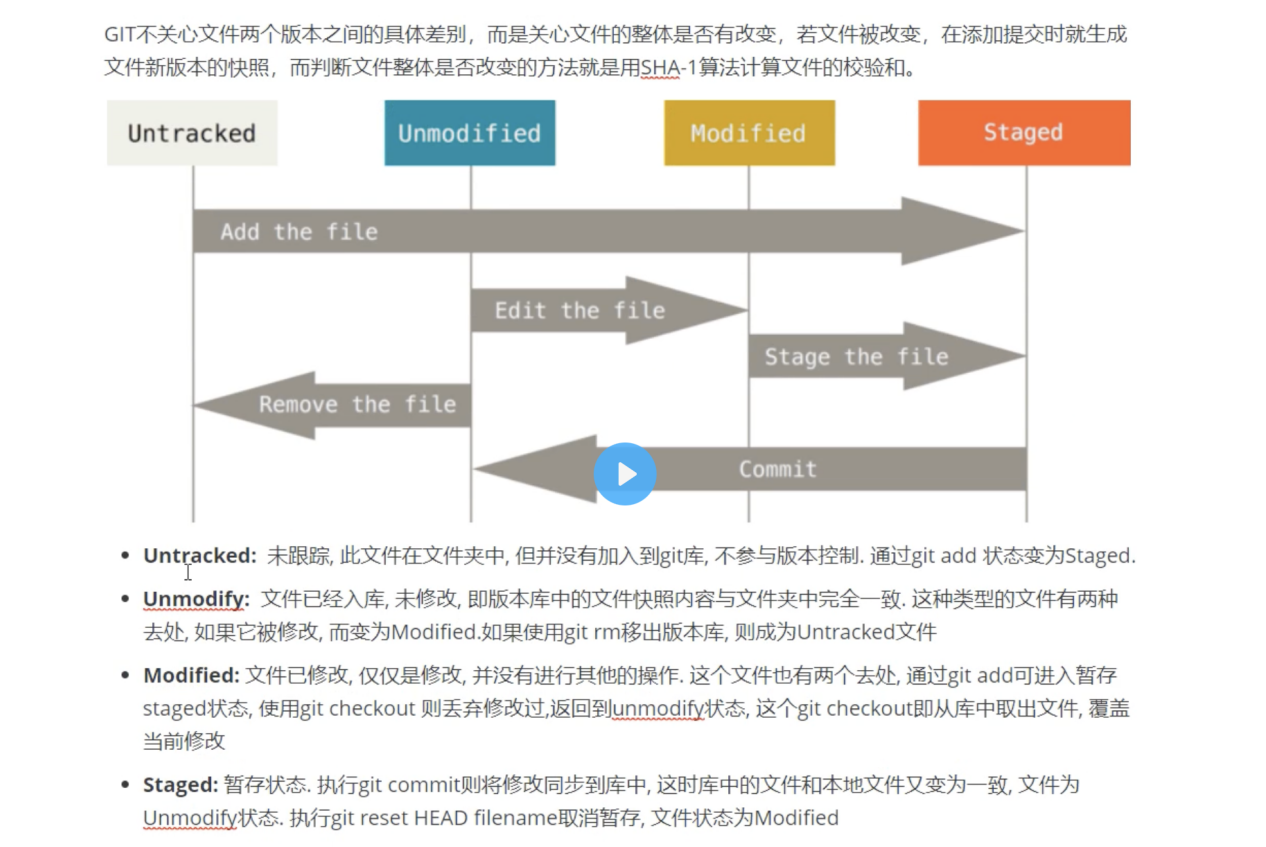
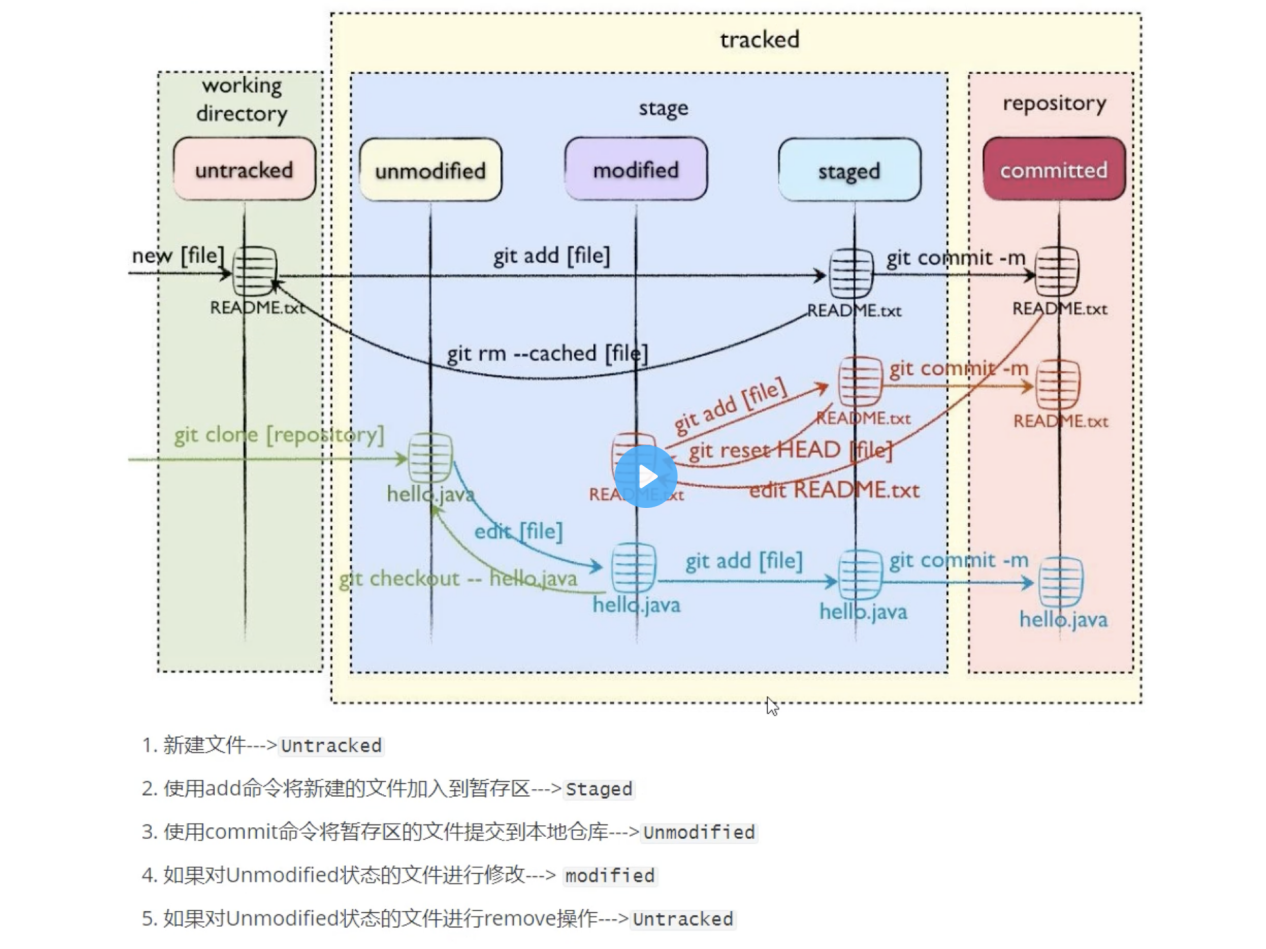
**git简介**

1. git的四个工作区域



1. git文件的四种状态





**git命令**

1. git常见命令

*git branch 查看本地所有分支*

*git status 查看当前状态*

*git commit 提交*

*git branch -a 查看所有的分支*

*git branch -r 查看远程所有分支*

*git commit -am "init" 提交并且加注释*

*git remote add origin git@192.168.1.119:ndshow*

*git push origin master 将文件给推到服务器上*

*git remote show origin 显示远程库origin里的资源*

*git push origin master:develop*

*git push origin master:hb-dev 将本地库与服务器上的库进行关联*

*git checkout --track origin/dev 切换到远程dev分支*

*git branch -D master develop 删除本地库develop*

*git checkout -b dev 建立一个新的本地分支dev*

*git merge origin/dev 将分支dev与当前分支进行合并*

*git checkout dev 切换到本地dev分支*

*git remote show 查看远程库*

*git add .*

*git rm 文件名(包括路径) 从git中删除指定文件*

*git clone git://github.com/schacon/grit.git 从服务器上将代码给拉下来*

*git config --list 看所有用户*

*git ls-files 看已经被提交的*

*git rm [file name] 删除一个文件*

*git commit -a 提交当前repos的所有的改变*

*git add [file name] 添加一个文件到git index*

*git commit -v 当你用－v参数的时候可以看commit的差异*

*git commit -m "This is the message describing the commit" 添加commit信息*

*git commit -a -a是代表add，把所有的change加到git index里然后再commit*

*git commit -a -v 一般提交命令*

*git log 看你commit的日志*

*git diff 查看尚未暂存的更新*

*git rm a.a 移除文件(从暂存区和工作区中删除)*

*git rm --cached a.a 移除文件(只从暂存区中删除)*

*git commit -m "remove" 移除文件(从Git中删除)*

*git rm -f a.a 强行移除修改后文件(从暂存区和工作区中删除)*

*git diff --cached 或 $ git diff --staged 查看尚未提交的更新*

*git stash push 将文件给push到一个临时空间中*

*git stash pop 将文件从临时空间pop下来*

*---------------------------------------------------------*

*git remote add origin git@github.com:username/Hello-World.git*

*git push origin master 将本地项目给提交到服务器中*

*-----------------------------------------------------------*

*git pull 本地与服务器端同步*

*-----------------------------------------------------------------*

*git push (远程仓库名) (分支名) 将本地分支推送到服务器上去。*

*git push origin serverfix:awesomebranch*

*------------------------------------------------------------------*

*git fetch 相当于是从远程获取最新版本到本地，不会自动merge*

*git commit -a -m "log\_message" (-a是提交所有改动，-m是加入log信息) 本地修改同步至服务器端 ：*

*git branch branch\_0.1 master 从主分支master创建branch\_0.1分支*

*git branch -m branch\_0.1 branch\_1.0 将branch\_0.1重命名为branch\_1.0*

*git checkout branch\_1.0/master 切换到branch\_1.0/master分支*

*du -hs*

*git branch 删除远程branch*

*git push origin :branch\_remote\_name*

*git branch -r -d branch\_remote\_name*

1. 配置用户签名

方式一：单个仓库有效

*git config user.name 用户名*

*git config user.email 邮箱*

方式二：全局有效

*git config --global user.name 用户名*

*git config --global user.email 邮箱*

一般情况下设置成全局有效即可，需要单独为某仓库配置信息时，再按方式一即可(优先级高，覆盖全局配置)

1. 初始化本地仓库

*git init*

*$>tree -L 1*

*.*

*|-- HEAD # 这个git项目当前处在哪个分支里*

*|-- config # 项目的配置信息，git config命令会改动它*

*|-- description # 项目的描述信息*

*|-- hooks/ # 系统默认钩子脚本目录*

*|-- index # 索引文件，暂存区（stage）*

*|-- logs/ # 各个refs的历史信息*

*|-- objects/ # Git本地仓库的所有对象 (commits, trees, blobs, tags)*

*|-- refs/ # 标识你项目里的每个分支指向了哪个提交(commit)。*

1. 查看git状态

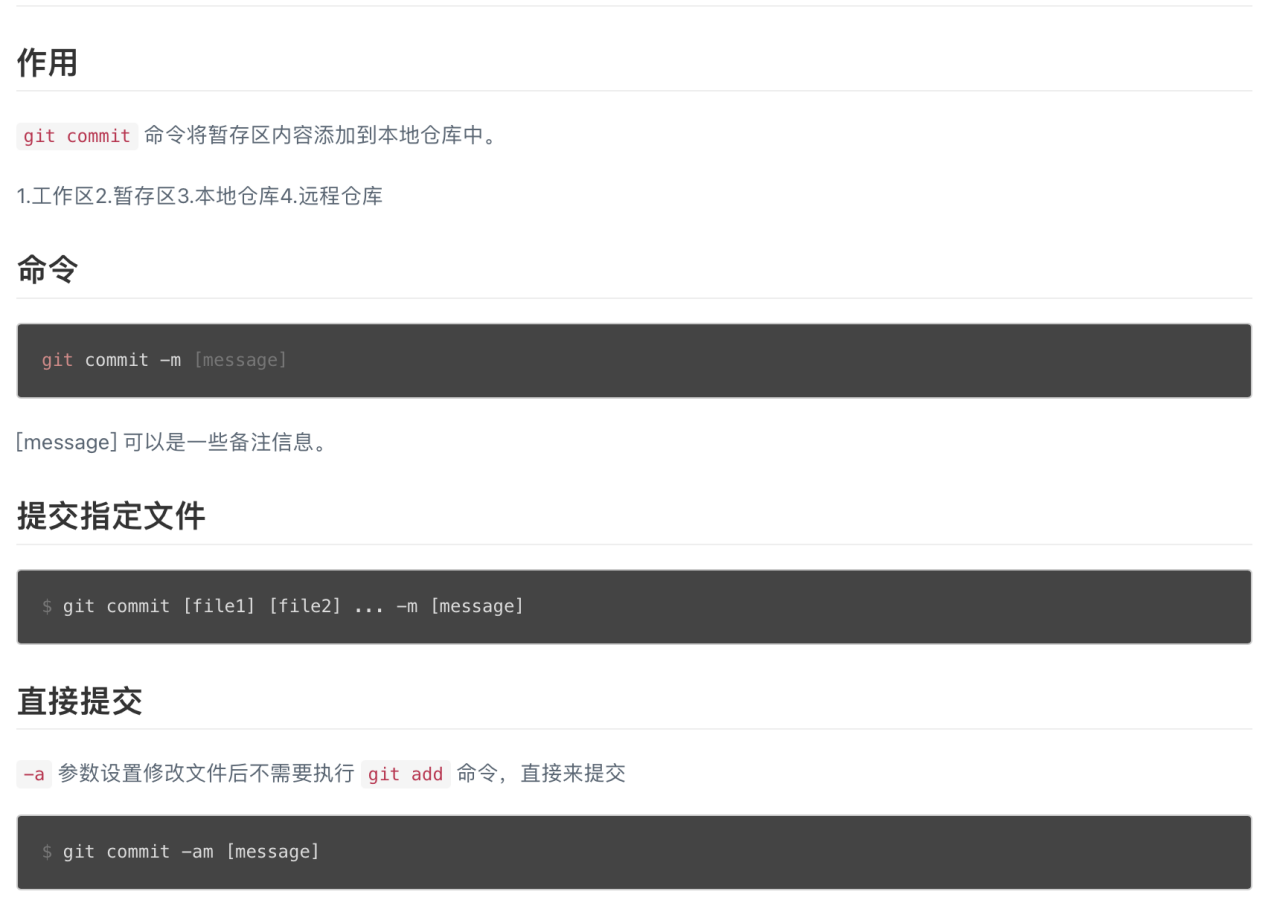
*git status*



1. 将代码添加到暂存区



1. 将代码提交到本地仓库



1. 修改并提交代码
2. 项目版本切换

*reflog 查看版本信息*

*git reset --hard 版本号 xxx*

**git分支**

1. git分支常用命令

查看分支

*git branch -v*

创建分支

*git branch 新分支名*

删除分支

*git branch -d branchname*

修改分支

*git branch -m oldbranchname newbranchname*

切换分支

*git checkout branchname*

合并分支

*git merge branchname*

1. 合并分支



1. 解决合并分支冲突

产生原因：两个分支在同一位置都有修改，git无法判断去留

解决方式：手动修改存在冲突的文件，然后commit -am，再merge

GitHub

1. Git团队协作
2. 创建远程库



1. 推送本地库

*git push 别名 分支*

1. 拉取远程库

*git pull 别名 分支*

1. 克隆远程库

*git clone url*

1. GitHub团队内协作



1. GitHub跨团队协作

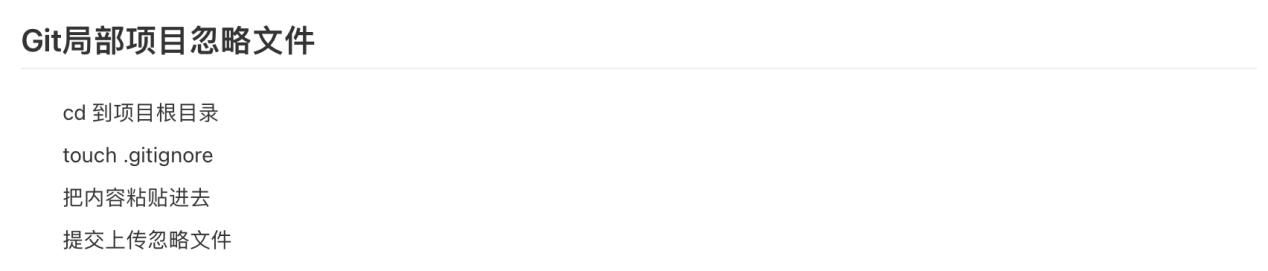


1. 免密码登录

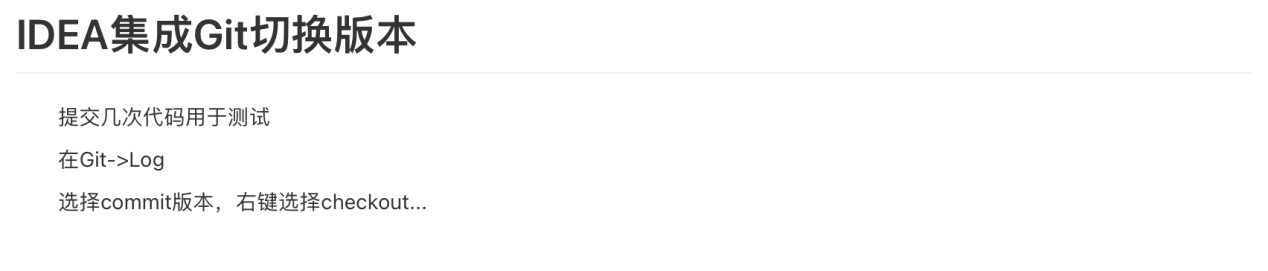


**IDEA集成Git**

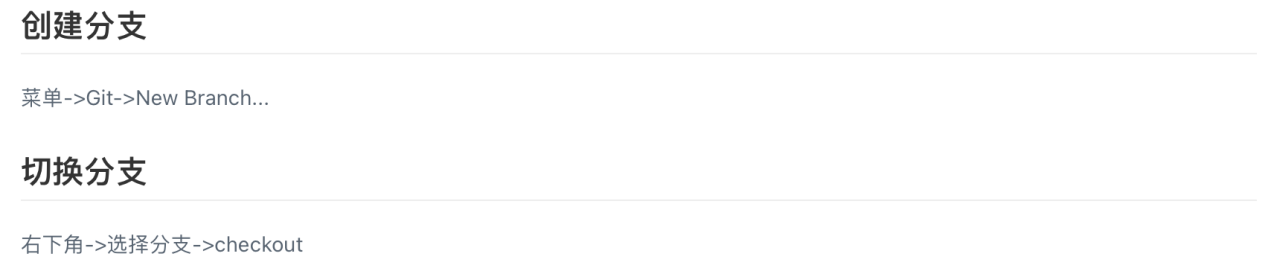
1. 配置git忽略文件



1. IDEA集成Git切换版本



1. IDEA集成Git创建和切换分支



1. IDEA集成Git合并分支

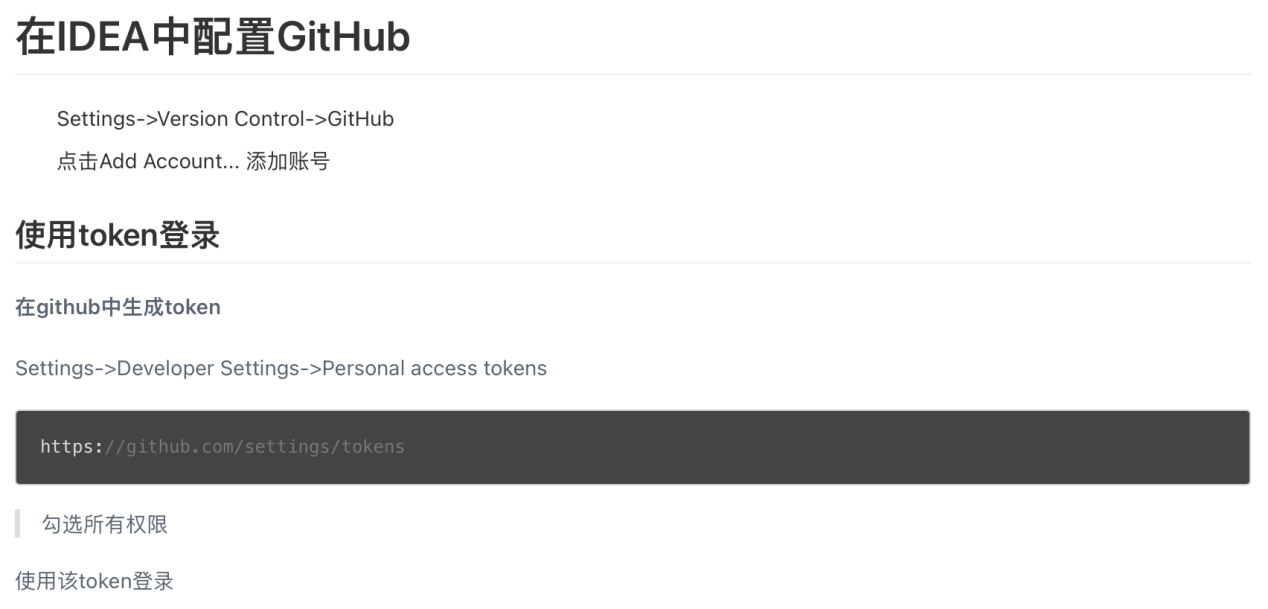


1. IDEA集成Git解决合并冲突

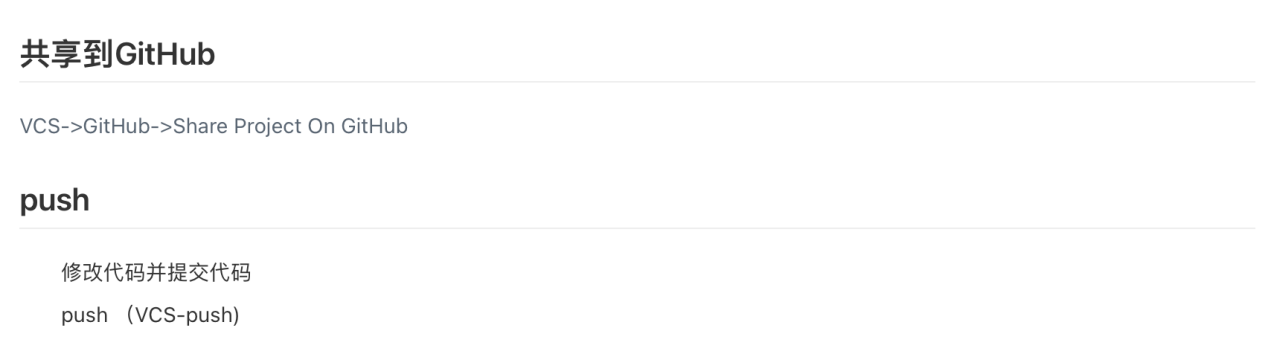
手动选择accept yours或者theirs

**IDEA集成GitHub**

1. 在IDEA中配置GitHub



1. 在IDEA中推送项目到GitHub



1. 在IDEA中从GitHub拉取项目

