# 框架高级课程系列之Git

## 尚硅谷JavaEE教研组

## 版本：V2.0

**课程计划-1天**

* Git概述
* Git安装、环境搭建
* Git命令行实战
* Github命令行实战
* IDEA中Github实战
* Git工作流
* Gitlab私服(了解)

# 第一章 概述

## 1.1 是什么

### 介绍

* + 源代码管理
  + Git是目前世界上最先进的分布式版本控制系统
  + Git诞生

计算机生成了可选文字:
LinUSTbrV日！ds
林纳斯。托瓦兹
一―\\
竺‘eep笠尸
免费用可以，
但是别破解

|  |
| --- |
| 在Linus Torvalds开发出了Git分布式版本控制系统11年后，BitKeeper宣布在Apache 2.0许可证下开源。Linus 是在BitKeeper停止向开源社区提供免费版本后开发了Git，今天Git和其它自由软件版本控制系统已经统治了编程社区，BitKeeper变得无关紧要了。BitKeeper由BitMover公司开发，该公司CEO Larry McVoy与Linus曾是好友， Larry说服Linus在内核开发中使用BitKeeper。而BitKeeper在免费使用的许可证中加入很多限制条件，惹恼了内核开发者，最终促使Linus开发出了毁灭BitMover的Git。 |

### 历史时间轴

计算机生成了可选文字:
Lin“系茂版本榨制历史
unuxh区£法像
公司堞对参与开璧
者行强有力的约
“本人手动合并代
2凶2巫
开发“孬的耔0“试图破
B“，的协议，
BM-'公司发墁，要收回
“社的兔費使用杖．
11年
2，年
“p“的东家
M““公司出于人
主文。授Lin“
区兔用这个版本冲
系垤．但玳不
行破，
L“自己用（言开发了一
个分布式版本控系
主体序廾发曦只用了周
一个月u系代由G理
@H勘上线'y

### 版本控制工具

* 集中式版本控制工具
  + - CVS、SVN(Subversion)、VSS……
* 分布式版本控制工具
  + - Git、Mercurial、Bazaar、Darcs……

## 1.2 能干嘛

### 作用

计算机生成了可选文字:
协同0
本堙
代记

### Git工作机制

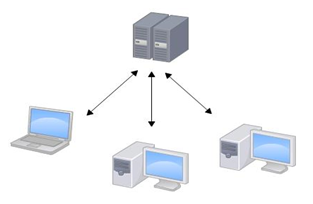
计算机生成了可选文字:
历史
版本
临时
存储
写代码
本地库
gitcommit
暂存区
gitadd
工作区

### 代码托管中心

代码托管中心负责维护远程库

* 局域网
  + - Gitlab
* Internet
  + - GitHub
    - 码云(gitee)

### 集中式版本控制系统



* SVN是集中式版本控制系统，版本库是集中放在中央服务器的，而干活的时候，用的都是自己的电脑，所以首先要从中央服务器下载最新的版本，然后开发代码，写完代码后需要把自己的代码推送到中央服务器。
* 缺点：
  + - 服务器单点故障
    - 容错性差

### 分布式版本控制系统

计算机生成了可选文字:
Fork
GitHub
PullRequest
push
Git
reflog
reset
clone
add
commit
push
add
commit
clone
Git
reflog
reset
push
add
commit
clone
reflog
reset
工作
发者A
工作
发者B
工作x
开发者x

* **Git**是分布式版本控制系统（Distributed Version Control System，简称 DVCS）
* **仓库：**
  + 本地仓库：是在开发人员自己电脑上的Git仓库
  + 远程仓库：是在远程服务器上的Git仓库
    - **团队内协作**
    - **跨团队协作**
* **操作：**
  + clone：克隆，就是将远程仓库复制到本地
  + push：推送，就是将本地仓库代码上传到远程仓库
  + pull：拉取，就是将远程仓库代码下载到本地仓库
  + Commit：提交，就是将本地工作区代码上传到本地仓库中
* **优点：**
  + 容灾能力强
  + 本地版本管理
  + 异地协作
  + 灵活分支

## 1.3 去哪下

### 命令行工具：Git for windows

下载地址：Git-2.23.0-64-bit.exe

<https://git-for-windows.github.io/>

<https://git-scm.com/download>

计算机生成了可选文字:
自9itfo阴indows.org
git，。rw'ndow'
VERS!ON2.23.0
FAQ
REPOSITORY
MA比INGLIST
Webringthe
aweSOme
itSCMtoWindows
DOWnload

### 操作系统中可视化工具：TortoiseGit(了解)

下载地址： <https://tortoisegit.org/>

TortoiseGit 2.8.0-64-bit

计算机生成了可选文字:
O自tortoisegit.org
TOft015eGit
WindowSShellInterfacetoGit
,,....................．叫口户．.r
About
Download
Supportv
Contribute
脚。
,
ThePowerofGit一
inaVVindowsshell
＼、一
\__
臂
岛
留目侧目困口月
、内j刁止
/'｝刁
洲厂口
产
“心舜飞
叹少哭乏～
匕
TOrtoiseGitprovidesoverlayiconsshowlngthe行leStatus
apewe而1conteXtmenuforGitandmuchmore!
Le己rnmo陀日b0UtTortoiseGit.J户．..
＼厂
／乙
/
r
、七止
NeWS
02019一07一201SecudtyvuLnerabiLjtjes石xedin匹IIY口，旦因丛因匡团幼望鱼坦血丛
型巡迪鱼no玛basedonPu下叮0.72
02019一02一28}ReLeasedTOrtoiseGit2.8.0
02018一08一191ReLeasedTOrtoiseGjt2.7.0
Re!eases嘴
0丛旦直鱼二2卫卫｝Reteasedon2019·02·28
0鱼哩叼竣业立王旦j}ReLeasedon2019一10·03

### GitHub网站

[http://www.github.com](http://www.github.com/)

计算机生成了可选文字:
自9ithub.com
奋｝q
☆。爪
WhyGitHub?,EnterpriseExploreVMarketplacePricing
519nin
卜―L口口口曰哭
、国勺
嘴
片日口口口口口曰曰
USeFname
Builtfor
developers
Email
paSSW0rd
GitHub15adevelopmentplafforminspiredbytheway
youwork.Fromopensourcetobusiness,youcanhost
andreviewcode,manageprojects,andbuildsoffware
alongside40milliondevelopers.
Makesureit'5atleast15charactersORatleast8charactersincludinga
nUmbeFanda10WerCase}etteF.LeaFnm0Fe.
5ignupforGitHub
声
厂
Byclicking"SignupforGitHub",youagreetoourTermsofServiceand
PrivacyStatementW甘11occasionallysendyouaccountrelatedemails.

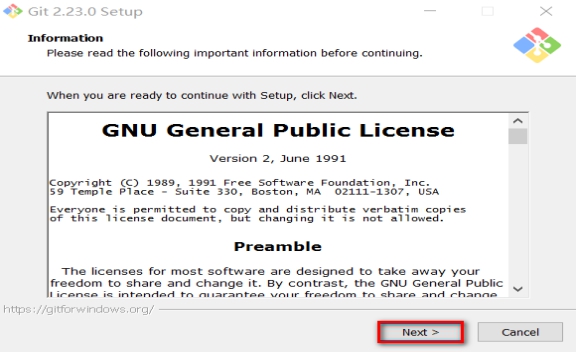
# 第二章 安装、环境搭建

## 2.1 安装

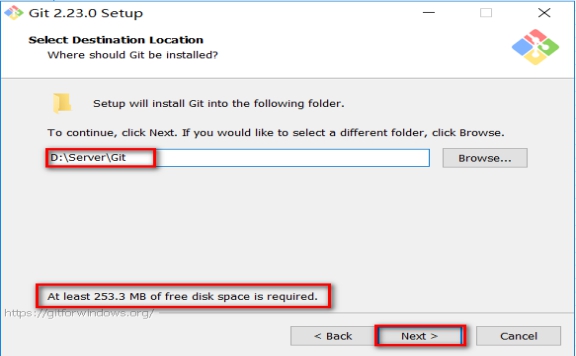
### 出现安全警告，点击运行



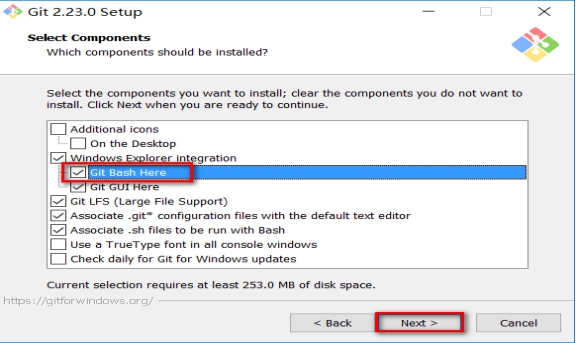
### 浏览协议，下一步



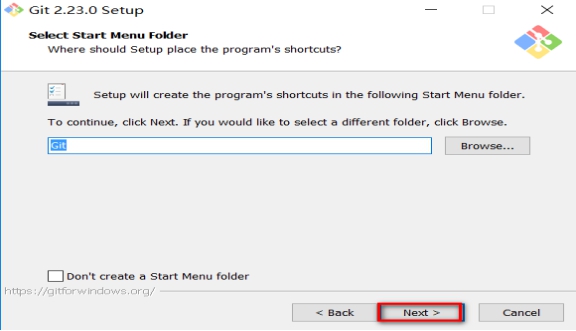
### 安装目录，所需要磁盘空间大小，下一步



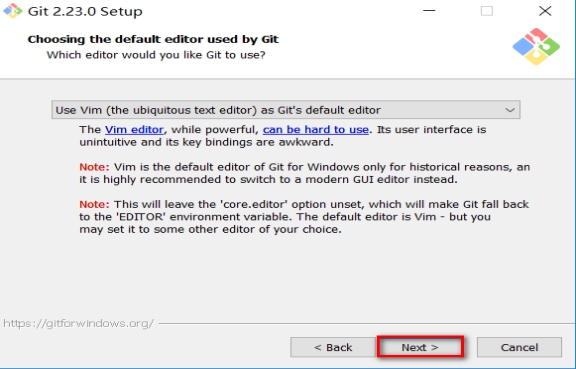
### Git Bash需要安装的，其他默认即可，下一步



### 开始菜单，下一步

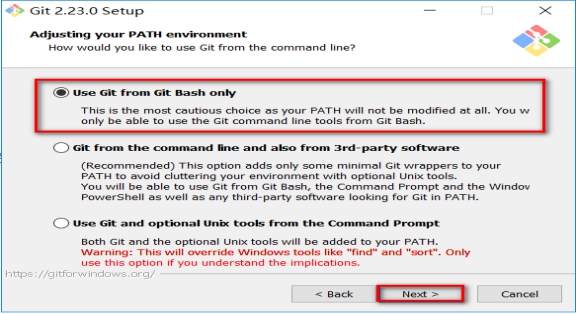


### 默认编辑器，下一步

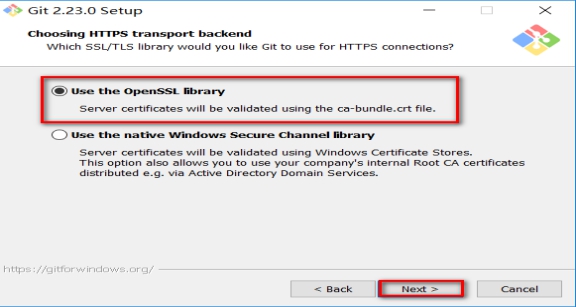


### 选择Git命令的执行环境

* 这里推荐选择第一个，就是单独用户Git自己的命令行窗口。
* 不推荐和windows的命令行窗口混用。



### HTTPS传输：使用SSL传输协议

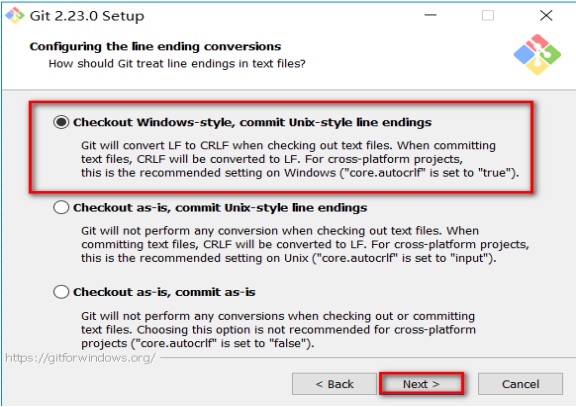


### 在“Configuring the line ending conversions”选项中，

第一个选项：如果是跨平台项目，在windows系统安装

第二个选项：如果是跨平台项目，在Unix系统安装

第三个选项：非跨平台项目



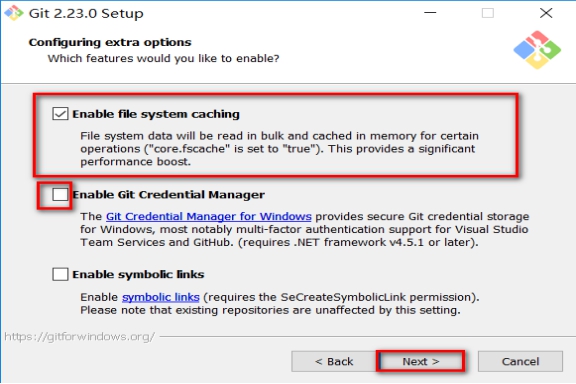
### 在“terminal emulator”选项中，

第一个选项：使用专用的Git窗口（**推荐**）

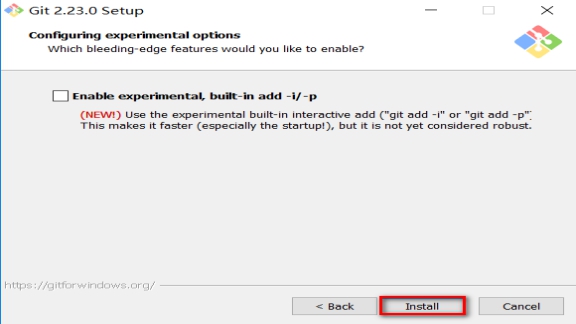
第二个选项：使用windows的cmd命令行窗口。



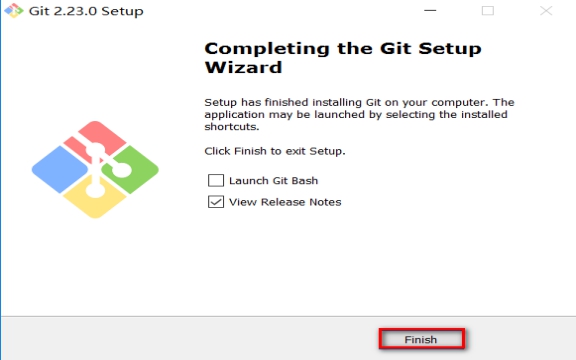
### 在“Configuring extra”选项中，默认开启文件缓存即可（推荐）



### 新功能，还不稳定，不勾选，开始安装



### 安装完成



## 2.2 设置全局用户签名

1. 安装完成后，在任意的文件目录下，右键都可以开打Git的命令行窗口——Git Bash Here

计算机生成了可选文字:
、／、／
查看凹
排序方式（0)
分组依据（P)
刷新（E)
>
自定义文件夹（日…
枯贴沙）
粘贴快捷方式（S)
箫肖移动（U)
GitGUIHere
GitBashHere
文件头劝口密
共享（H)
创svNche。kout…
弓TortoisesvN
新建即门
属性（R)
>
>

1. Git是分布式版本控制系统，所以需要填写用户名和邮箱作为一个标识——即：用户签名

* 说明
  + - 签名的作用是区分不同操作者身份。用户的签名信息在每一个版本的提交信息中能够看到，以此确认本次提交是谁做的。
  + 注意：这里设置用户签名和将来登录GitHub（或其他代码托管中心）的账号没有任何关系。
* 命令：

git config --global user.name "用户名"

git config --global user.email "用户邮箱"

* --global 表示全局属性，所有的git项目都会共用属性
* 查看配置信息：git config --list
* 在当前登录的系统用户路径下，生成~/.gitconfig隐含文件，里面可以看到刚刚设置的信息。如果不用命令设置，也可以直接打开文件进行设置。

# 第三章 相关概念

## 3.1 理解工作区、版本库、暂存区概念

* **工作区(Working Directory)**：就是你电脑本地硬盘目录，一般是项目当前目录
* **版本库(Repository)**：工作区有个隐藏目录.git，它就是Git的本地版本库
* **暂存区(stage)**：一般存放在"git目录"下的index文件（.git/index）中，所以我们把暂存区有时也叫作索引（index）
* **分支（Branch）**：Git为我们自动创建的第一个分支master，以及指向master的一个指针叫HEAD

计算机生成了可选文字:
工作区
9it版本库H
香存曰
D
add
本地库
DDD
DDD
CheCk0UtL一＜file
叫卜～
COmmit
Q
A
E
―日「
Feset一haFd
叫卜－
删除文件
．仁卜
曼
COmmit
add【＜filenam且＞
.
l

## 3.2 提交Git版本库分两步执行

**第一步** 用“git add”把文件纳入Git管理，实际是把本地文件修改添加到暂存区

**第二步** 用“git commit”提交更改，实际上就是把暂存区的所有内容提交到当前分支

因为我们创建Git版本库时，Git自动为我们创建了唯一一个master分支，所以commit就是往master分支上提交更改。

可以简单理解为，需要提交的文件修改通通放到暂存区，然后，一次性提交暂存区的所有修改。一旦提交完后，如果你又没有对工作区做任何修改，那么工作区就是“干净”的。即：nothing to commit (working directory clean)。

**其他操作**

①用“git diff HEAD -- filename”命令可以查看工作区和暂存区里面最新版本的区别。

②新建过撤销未add： git checkout -- 文件名

③撤销已add未commit：先git reset [HEAD] 文件名，再 git checkout -- 文件名

④撤销已add已commit：git reset --hard HEAD^

# 第四章 Git本地库实战

* 常用命令预览：

|  |  |
| --- | --- |
| 命令名称 | 命令作用 |
| git init | 初始化本地库 |
| git config --global user.name 用户名 | 设置用户签名的用户名部分 |
| git config --global user.email 邮箱 | 设置用户签名的邮箱部分 |
| git status | 查看本地库状态 |
| git add 文件名 | 添加到暂存区 |
| git commit -m "日志信息" 文件名 | 提交到本地库 |
| git reflog | 查看历史记录 |
| git reset --hard 版本号 | 版本穿梭 |

## 4.1 实战(初始化版本库)

* 要使用Git对我们的代码进行版本控制，首先需要获得Git仓库，获取Git仓库通常有两种方式：
  + - 在本地初始化一个Git仓库
    - 从远程仓库克隆
* 本地初始化操作步骤：
  + - 创建目录（用作本地版本库），例如：D:\DevRepository\GitRepository\oa，oa表示办公自动化项目名称
    - 当前目录打开Git Bash窗口，初始化仓库

命令：git init

* + - 查看当前目录产生.git隐藏文件夹
    - 注意：当前的 .git 是一个隐藏文件夹，需要在电脑上面设置隐藏文件夹显示设置，否则，看不到这个.git目录

计算机生成了可选文字:
超大标过大标
小标列表
二内容
是中标
囗分组
添加列
悱序方式
将所有列调整为台适的大小
囗顶目3框
文亻牛扩展名
9]的顶所选顶

## 4.2 实战(新建\提交\状态)

1. 新建文件：
   * 命令：touch a.txt
   * 命令：vim a.txt
2. 查看文件状态
   * 命令：
     + **git status**
       - On branch master ：表示主分支
       - Untracked files：表示未跟踪状态
   * 说明：
     + Git工作目录下的文件状态信息：
       - Untracked 未跟踪（未被纳入版本控制）
       - Tracked 已跟踪（被纳入版本控制）
       - Unmodified 未修改状态
       - Modified 已修改状态
       - Staged 已暂存状态
     + 这些文件的状态会随着我们执行Git的命令发生变化
       - 红色表示新建文件或者新修改的文件,都在工作区.
       - 绿色表示文件在暂存区
       - 新建的文件在工作区，需要添加到暂存区并提交到仓库区
     + 也可以使用git status –s 使输出信息更加简洁
       - ?? : 表示未跟踪状态
3. 添加到暂存区
   * 命令：
     + **git add** <文件名称>
       - 只是增加到栈空间（index文件）中，还没有添加到本地库中。初始化时没有这个index文件。这还是一个新文件，需要将栈空间文件提交到本地仓库。
     + **git add .**
       - 添加项目中所有文件
     + **git add hello.txt**
       - 添加未存在文件会出错：fatal: pathspec 'hello.txt' did not match any files
4. 撤销暂存区的文件
   * 命令：
     + **git reset <文件名称>**
     + 撤销后，查看文件状态（git status）文件由绿色变为红色
5. 将暂存区文件提交到本地库
   * 命令：
     + git commit
       - 执行命令时需要填写提交日志，进入编辑模式
     + git commit –m “注释内容”
       - 直接用-m参数指定日志内容，推荐
       - commit 会生成一条版本记录，add只是添加暂存区，不会生成版本记录，建议多次add后，一次性commit，避免每次add都commit产生版本信息爆炸。
     + git commit -am "注释内容"
       - 代码编辑完成后即可进行 add 和 commit 操作
       - 提示：添加和提交合并命令
6. 编辑保存再提交，反复练习多次，修改提交3次(为接下来实验做准备)

## 4.3 实战(查看日志)

* 命令：
  + git log
  + **git log a.txt**
    - 查看文件日志(查看所有日志或某个文件日志)
    - q退出
  + **git log --pretty=oneline**
    - 如果日志很多,可以在一行显示
  + **git reflog**
    - 查看历史操作

## 4.4 实战(回退\穿梭\撤销)

1. 回退到历史版本
   * 命令：
     + git reset --hard HEAD^
       - 一次回退一个版本，一个^代表一个版本数量
     + git reset --hard HEAD~n
       - 回退n次操作
2. 版本穿梭
   * 命令：
     + git reflog a.txt
       - 查看历史操作
     + git reset --hard 版本号
       - 回到最新的版本
3. 撤销：
   * 未add，未commit
     + vim修改文件，没有add和commit，进行撤销
     + 命令：
       - git checkout -- a.txt
         * 撤销修改(还原原来的文件)
   * 已add，未commit
     + vim修改文件，添加add，但没提交commit，进行撤销
     + 命令：
       - git add a.txt
       - **git reset**
         * 软回退：
         * 查看文件内容：cat a.txt
         * 查看日志：git reflog a.txt

## 4.5 实战(删除)

1. 手动拷贝图片java.jpg到工作空间目录，并查看目录列表：ls -l
2. 添加：git add java.jpg
3. 提交：git commit -m "新建图片" java.jpg
4. 删除图片：rm java.jpg
5. 添加：git add java.jpg
6. 提交：git commit -m "新建图片" java.jpg
7. 回退：git reset --hard HEAD^
8. 文件不是被删除了吗？怎么又回来啦！呵呵…
9. 处处留痕：git reflog

## 4.6 实战(分支)

几乎所有的版本控制系统都以某种形式支持分支。 使用分支意味着你可以把你的工作从开发主线上分离开来，以免影响开发主线。Git 的master分支并不是一个特殊分支。 它跟其它分支没有区别。 之所以几乎每一个仓库都有 master 分支，是因为git init 命令默认创建它，并且大多数人都懒得去改动它。

### 实际开发流程

计算机生成了可选文字:
重亲斤
讠殳十
产品经理
内部
不通过专
评审
通过
需求
评审
通过
需求排期
分配任务
产品原型
开发人员参与
Bug
服务器
模块之间
相互调用
廾发
服务器
O冂0
开发工程师
版本
测试
服务器
《测试工程师》
反馈
最终用户
服务器

### 分支概念

在使用版本控制工具开发的过程中，同时推进多个任务

计算机生成了可选文字:
hot
master
feature—blue
feature—game

### 分支的好处

同时并行推进多个功能开发，提高开发效率

各个分支在开发过程中，如果某一个分支开发失败，不会对其他分支有任何影响。失败的分支删除重新开始即可。

### 分支操作

|  |  |
| --- | --- |
| 命令名称 | 作用 |
| git branch 分支名 | 创建分支 |
| git branch -v | 查看分支 |
| git checkout 分支名 | 切换分支 |
| git merge 分支名 | 把指定的分支合并到当前分支上 |

### 分支操作详解

#### 创建分支和切换分支图解

计算机生成了可选文字:
第一次
第二次
master
第三次
HEAD
hotfix
第四次

master、hotfix其实都是指向具体版本记录的指针。当前所在的分支，其实是由HEAD决定的。

HEAD如果指向master，那么我们现在就在master分支上。

HEAD如果指向hotfix，那么我们现在就在hotfix分支上。

所以切换分支的**本质就是移动HEAD指针**。

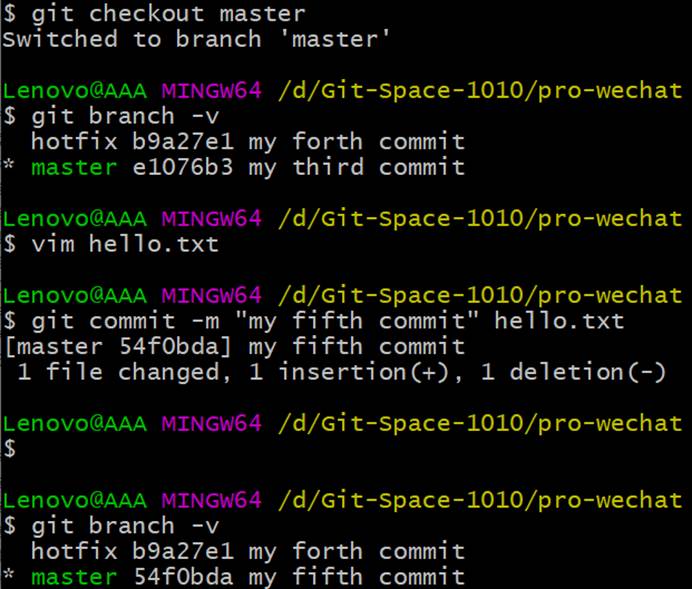
#### 创建分支和切换分支的操作截图



#### 在两个不同分支分别做不同修改图解

计算机生成了可选文字:
第二次
hotfix
第四次
第五次
master
HEAD

#### 在两个不同分支分别做不同修改操作截图

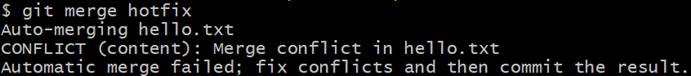


#### 合并分支图解

计算机生成了可选文字:
第一次
第二次
hotfix
第四次
merge
第六次
master
1、
HEAD

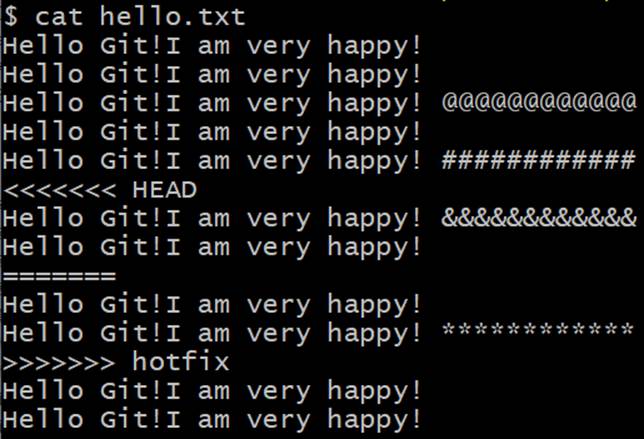
* 合并分支时一定是涉及到两个分支。这两个分支一个是“当前所在分支”，一个是“目标分支”。
* 命令写法：
  + - git merge 目标分支
    - 所以分支合并命令的本质就是把“目标分支”合并到“当前分支”。
* 例如：把hotfix合并到master
  + - git merge hotfx
    - 需要确保当前所在的分支是master
* 例如：把master合并到hotfix
  + - git merge master
    - 需要确保当前所在的分支是hotfix

### 合并分支操作截图



## 4.7 实战(版本冲突)

### 冲突产生的表现



Git使用“<<<<<<<、=========、>>>>>>>>>>”符号帮我们标记出来，现在产生冲突的内容。

|  |
| --- |
| <<<<<<< HEAD  Hello Git!I am very happy! &&&&&&&&&&&&  Hello Git!I am very happy!  =======  表示HEAD指针指向的位置在冲突中的内容 |

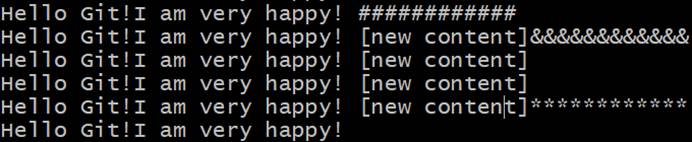
|  |
| --- |
| =======  Hello Git!I am very happy!  Hello Git!I am very happy! \*\*\*\*\*\*\*\*\*\*\*\*  >>>>>>> hotfix  表示hotfix指针指向的位置在冲突中的内容 |

### 冲突产生的原因

合并分支时，两个分支在同一个文件的同一个位置有两套完全不同的修改。Git无法替我们决定使用哪一个。必须人为决定新代码内容。

### 冲突的解决

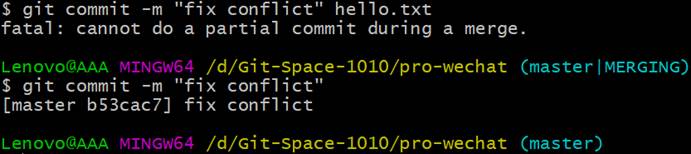
①编辑有冲突的文件，删除特殊符号，决定要使用的内容



②添加到暂存区



③执行提交（注意：使用git commit命令时不能带文件名）



### 避免冲突

* 容易冲突的操作方式
  + - 多个人同时操作了同一个文件
    - 一个人一直写不提交
    - 修改之前不更新最新代码
    - 提交之前不更新最新代码
    - 擅自修改同事代码
* 减少冲突的操作方式
  + - 养成良好的操作习惯，先`pull`在修改,修改完立即`commit`和`push`
    - 一定要确保自己正在修改的文件是最新版本的
    - 各自开发各自的模块
    - 如果要修改公共文件，一定要先确认有没有人正在修改
    - 下班前一定要提交代码,上班第一件事拉取最新代码
    - 一定不要擅自修改同事的代码

# 第五章 GitHub远程库实战

## 5.1 Git代码托管服务

前面我们已经知道了Git中存在两种类型的仓库，即**本地仓库**和**远程仓库**。那么我们如何搭建Git远程仓库呢？我们可以借助互联网上提供的一些代码托管服务来实现，其中比较常用的有GitHub、码云、GitLab等。

* gitHub（ 地址：<https://github.com/> ）

是一个面向开源及私有软件项目的托管平台，因为只支持Git 作为唯一的版本库格式进行托管，故名gitHub

* 码云（地址： <https://gitee.com/> ）

是**国内**的一个代码托管平台，由于服务器在国内，所以相比于GitHub，码云速度会更快

* GitLab （地址： <https://about.gitlab.com/> ）

是一个用于仓库管理系统的开源项目，使用Git作为代码管理工具，并在此基础上搭建起来的web服务

## 5.2 Github简介

1.是什么： github是一个git项目托管网站，主要提供基于git的版本托管服务

2.能干嘛： github是一个基于git的代码托管平台， Git 并不像 SVN 那样有个中心服务器。目前我们使用到的 Git 命令都是在本地执行，如果你想通过 Git 分享你的代码或者与其他开发人员合作。 你就需要将数据放到一台其他开发人员能够连接的服务器上。

计算机生成了可选文字:
GitHub
（、l少：、
国扭
口口
Gi未

3.去哪下： <https://github.com/>

4.怎么玩：见课堂演示

前面执行的命令操作都是针对的本地仓库，本章节我们会学习关于远程仓库的一些操作，具体包括：

* 查看远程仓库
* 添加远程仓库
* 从远程仓库克隆
* 移除无效的远程仓库
* 从远程仓库中抓取与拉取
* 推送到远程仓库

## 5.3 Github实战—团队内协作

* 一共9个步骤：我们完全按照9个步骤的流程来进行练习。做到熟练掌握。

计算机生成了可选文字:
GitHUb
推送代码到远端一
1、9itremoteaddorigin<ur
2、9itpushoriginmaster
.
昌19,,p二”]
9itClone
<Url>
、9itaddx关关
、9itcommit
432
搭建代码库
9itCon朽g
9itcon石g
USer.name
USeF.email
口口

### [zhangsanatguigu]创建本地项目

/d/DevRepository/GITRepository/zhangsanatguigu

* + $ mkdir zhangsanatguigu
  + $ cd zhangsanatguigu
  + $ git init
  + $ git config user.name "zhangsanatguigu"
  + $ git config user.email "dev111\_atguigu@163.com"
  + D:\DevRepository\GITRepository\zhangsanatguigu\.git\config文件

计算机生成了可选文字:
[user〕
name二zhangsanatguigu
ema11二devlll_atguigu@163.com

### 提交代码到本地库

* + $ touch love.txt
  + $ vim love.txt
  + $ git add love.txt
  + $ git commit -m "love form zhangsanatguigu" love.txt

### [zhangsanatguigu]登录账号到github,创建远程仓库.

①登录后没有任何项目情况下,会显示如下页面



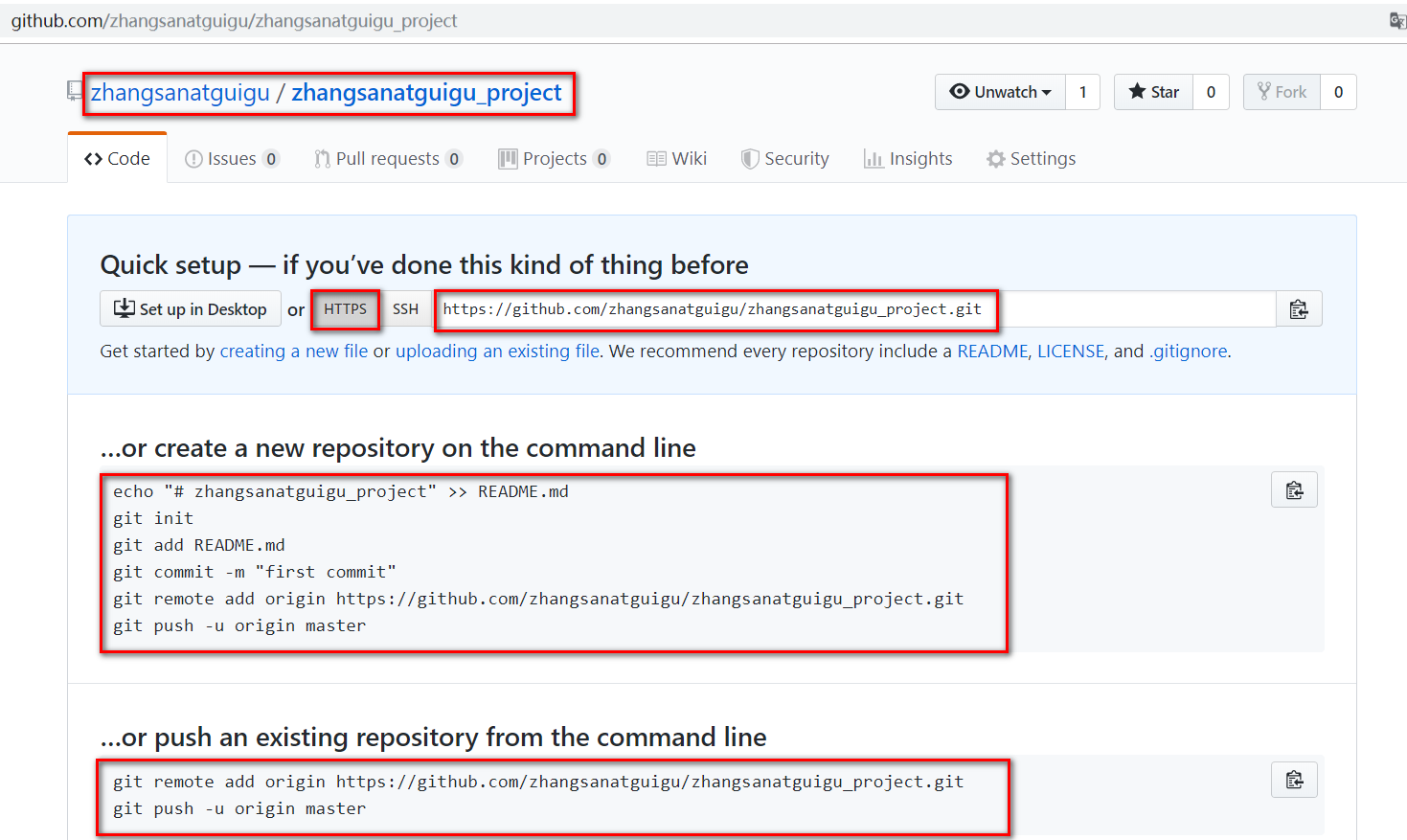
②创建新的仓库

计算机生成了可选文字:
坐型竺卫鲤竺曰
lmportrepository
Newgist
Neworganization

③填写创建新仓库信息

计算机生成了可选文字:
.github.com/new
Createanewrepository
Arepositorycontainsallprojectfiles,includingtherevisionhistory.Alreadyhaveaprojectrepositoryelsewhere?
lmportarepository.
OWneF
Repositoryname食
粤zhangsanatguigu!
了
Greatrepositorynamesareshortandmemorable.Needinspiration?Howaboutcautious一robot?
Description(optional)
pUbliC
Anyonecollseethisrepository.YOuchoosewhocancommit.
军一凸
口
pFIVate
YOuchoosewhocanseeandcommittothisrepository.
Skipthisstepifyou'reimportinganexistingrepository.
InitializethisrepositorywithaREADME
Thiswillletyouimmediatelyclonetherepositorytoyourcomputer.
Add.gitignore二None甲
Addalicense:None，①
Createrepository

④创建后出现仓库地址，注意：点一下HTTPS，换下地址。



### [zhangsanatguigu]本地项目推送到远程库

* + [zhangsanatguigu]增加远程地址
  + git remote add <远端代号> <远端地址>
  + <远端代号> 是指远程连接的代号,一般直接用origin作为代号,也可以自定义.
  + <远端地址> 默认远程连接的url
    - 例如：

git remote add origin <https://github.com/zhangsanatguigu/zhangsanatguigu_project.git>

计算机生成了可选文字:
圈。：\oevRepository\GI丁Repository\zhangsanatguigu\.git\con6g一Notepad++
口
X
月一
文件（F)
O一
编辑（日
、
搜索（S）视图（V）格式（M）语言（L）设置m宏（0)
。．。目…·汹巳｛寻、‘品乙渔｝滋之卜
运行（R）相副叙P）窗D(W)?
气葺〕图目｝困回困回后…．
百。。nfig司
[core〕
repositoryformatversion二o
filemode二fa1Se
bare二fa1Se
IOgallrefupdates二true
syml1nks二false
ignorecase二true
[user]
name
emal
zhangsanatguigu
devlll_at只ui只u@163.com
origin"]
1'."
「remote
Url
fetch二＋refs/heads/*:refs/remotes/origin/*
1234L卜678901234
11111
Norm日！teXt6le
length:326lines:14Ln:12Col:73Sel:0{0
UNIX
UTF一8w/oBOMINS

* git push <远端代号> <本地分支名称>
  + <远端代号> 是指远程链接的代号
  + <本地分支> 是指要提交的分支名字,比如master
  + 例如：git push origin master

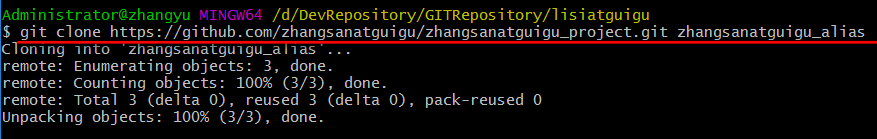


计算机生成了可选文字:
C0.github.com//11。。1195"11"1(]""',,I/,，『，',,j、·，『ld｝。j'J"]
翻q
☆！旦！
甲zhangsanatguigu
0Unwatch甲
★Staro
①Issueso九pullrequestso画Projectso
困W1ki困security匝Insights今settings
练习项目
Managetopics
Edit
01commit
驴1branch
令0releaseS
从1contributor
BranCh:master,
Newpullrequest
Createnew石leUpload石les
「．ndf:e「c.oneordown.oad!
粤zh。ngs。natguigul。vefrom：卜。ngs。n。tguigu
Latestcommitcf47f7344minutesago
国love.tXt
Iovefromzhangsanatguigu
44m1nutesago
HelppeopleinterestedinthisrepositoryunderstandyourprojectbyaddingaREADME.

### [lisiatguigu]从GitHub上克隆一个项目

* + 在D:\DevRepository\GITRepository\lisiatguigu目录下打开客户端窗口,执行命令下载远端项目
  + git clone <远端地址> <新项目目录名>。
    - <远端地址> 是指远程链接的地址。
    - <项目目录名> 是指为克隆的项目在本地新建的目录名称，可以不填，默认是GitHub的项目名。
    - 命令执行完后，会自动为这个远端地址建一个名为origin的代号。
  + 例如

git clone <https://github.com/zhangsanatguigu/zhangsanatguigu_project.git> hello\_world



### 使用[lisiatguigu]账户提交修改项目内容.

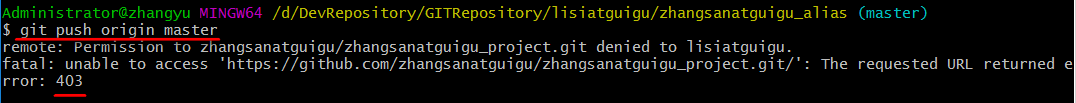
* + 配置账号和邮箱

计算机生成了可选文字:
Administrator公zhangyu侧64/d/DevRepository/'ITRepository/lisiatguigu/zhangsanatguigu_alias(master)
5gitconfiguser.name"lisiatguigu"
Administrator退zhangyuMTN翩64/d/DevRepository/'ITRepository/lisiatguigu/zhangsanatguigu_alias(master)
5gitconfiguser.email"dev222_atguigu尽163.com"

### 修改提交到本地库

计算机生成了可选文字:
Administrator公zhangyu·‘I闪‘八／d/DevRepository/'ITRepository/lisiatguigu/zhangsanatguigu_alias(master)
5vimlove.txt
Administrator公zhangyu’、”I闪G功·厂二／d/DevRepository/'ITRepository门isiatguigu/zhangsanatguigu一alias(master)
5gitaddlove.tXt
Administrator公zhangyuTN侧／d/DevRepository/'ITRepository门isiatguigu/zhangsanatguigu_alias(master)
5gitcommit一m"1151repyes',
[master4210fc8]1151repyes
1filechanged,1insertion(+)

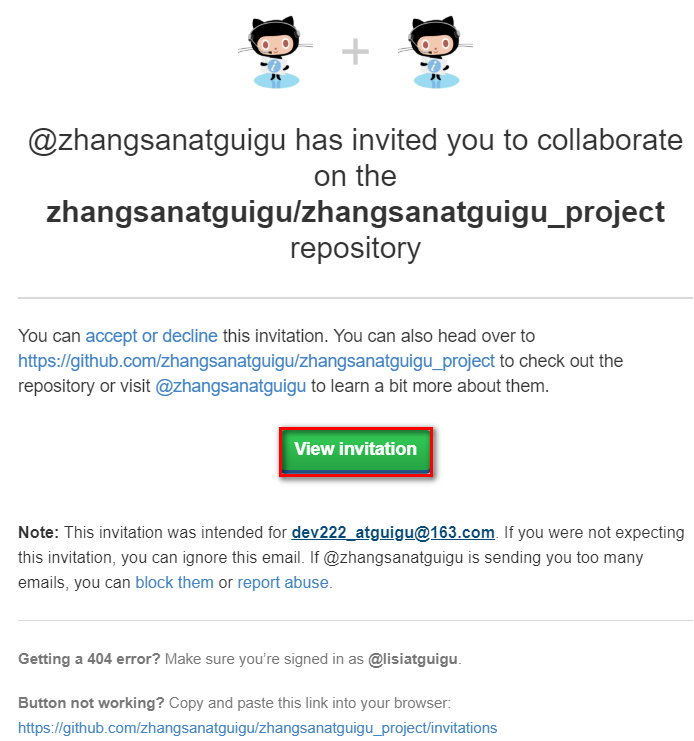
### 本地库提交到远程库，权限不够



1. 注:
   * clone会自动关联远端地址为origin默认名称,所以不需要git remote …
   * lisiatguigu账户提交修改内容,访问被拒绝.权限不够.
   * 以上对项目的操作方式，必须是项目的创建者或者合作伙伴。
   * 合作伙伴添加方式如下图: 在项目中点击settings页签，然后点击Collaborators,然后在文本框中搜索合作伙伴的邮箱或者账号。点击添加。
   * 添加后GitHub会给合作伙伴对应的邮箱发一封，邀请邮件。
2. [zhangsanatguigu]设置权限
   * Settings -> Collaborators -> [zhangsanatguigu]添加合作伙伴[lisiatguigu]

计算机生成了可选文字:
9ithUb.com/zhangsanatguigu/zhangsanatguigu_project/settingS/collaboration
O,
npto…
/PullrequestsIssuesMarketplaceEXplore
公Unwatch,1★Staro
？「orko
<>Code①lssueso
腌Pullrequestso
画projectso困wiki嘟secu「ity匝Insight、
optionS
CollaboratorS
Pushaccesstotherepository
Thisrepositorydoesn'thaveanycollaboratorsyet.Usetheformbelowtoaddacollaborator.
BFanCheS
Webh00ks
NotifiCationS
Searchbyusername,fullnameoremailaddress
YOu'11onlybeabletofindaGitHubuserbytheiremailaddressifthey'vechosentoIistitpublicly.othe阴ise,usetheirusername
instead.
Integrations&services
AddCollab0Fator
Deploykeys

* [zhangsanatguigu]邀请函已经发送到[lisiatguigu]邮箱
* [lisiatguigu]登录邮箱，打开链接



* [lisiatguigu]接受邀请,则合伙伙伴正式加入项目，获得直接提交代码的权限

计算机生成了可选文字:
9ithub.com//11厂，1'
．户上口二，
PullrequestslssuesMarketplaceEXplore
目zhangsanatguigu/zhangsanatguigu_Project
曰口口．训‘
匕口口口．电
目口口：,
zhangsanatguiguinvitedyoutocollaborate
Acceptinvitstion
DeCline
凸ownersofzhangsanatguigu_projectwillbea匕Ieto
See:
.YOurpublicprofileinformation
.Certainactivitywithinthisrepository
.Countryofrequestorigin
.YOuraccesslevelforthisrepository
.YOurlPaddress
15thisusersendingspamormaliciouscontent?
Blockzhangsanatguigu

1. [lisiatguigu]重新提交,OK



### [zhangsanatguigu]更新本地文件

* + 更新前查看

计算机生成了可选文字:
Administrator公zhangyu’一工闪‘、，一／d/oevRepository/'i丁Repository/zhangsanatguigu(master)
511
totall
一rw一r一r一1Administrator1971211110月2517:28love.txt
Administrator公zhangyu’·’工‘三，、／d/oevRepository/'I丁Repository/zhangsanatguigu(master)
cat1ove.txt
LOVEYOU
亡Jl

* 更新



* 更新后查看

计算机生成了可选文字:
Administrator公zhangyuMIN侧64/d/DevRepository/Gl下Repository/zhangsanatguigu(master)
cat1ove.txt
LOVEYOU
亡Jl
yes

#### 5.4 Github实战—团队内协作（协作冲突演示）

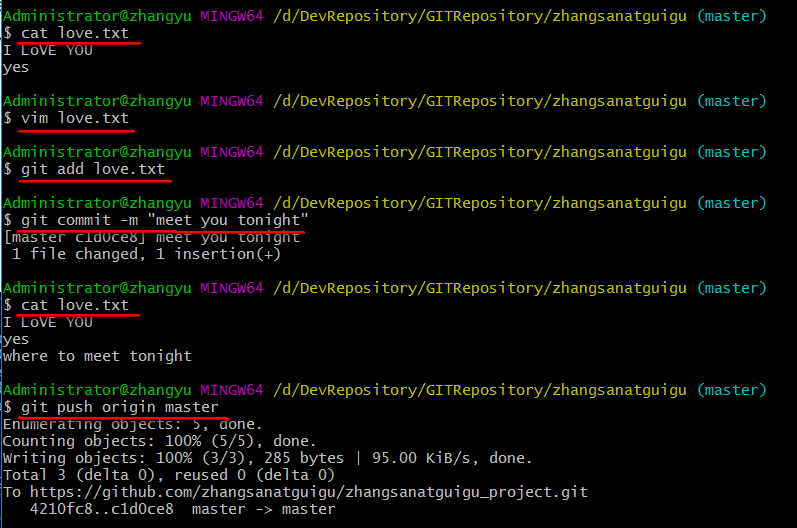
### 1.协作冲突

在上传或同步代码时，由于你和他人都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。

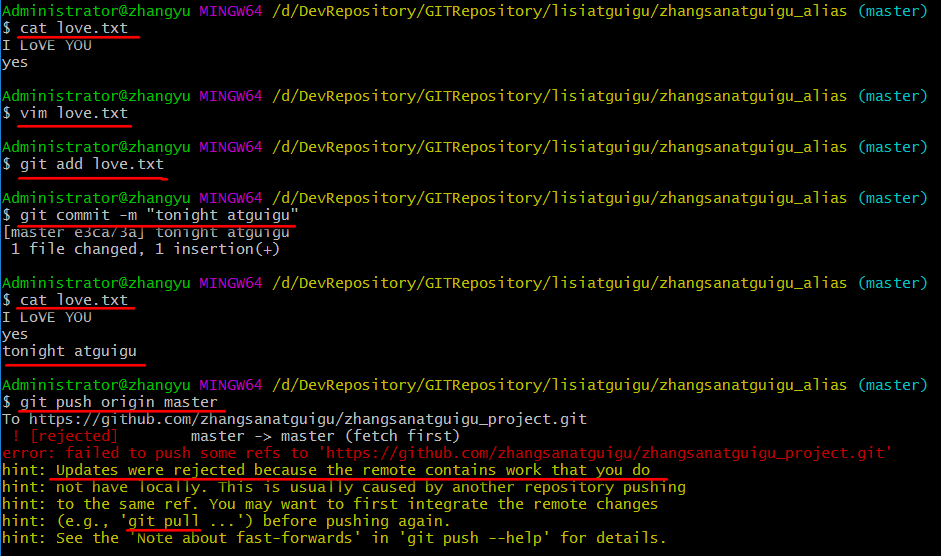
计算机生成了可选文字:
GitHUb
pull
侧口口口口口口口口口口口口口口口口口曰口口口晌
111
,~~~~~～叫卜
COnfliCt!
master
L
口口口口口口口口口口口口口口口口口口
push
push
master
m日SteF
成员A
成员B

### 2.演示冲突

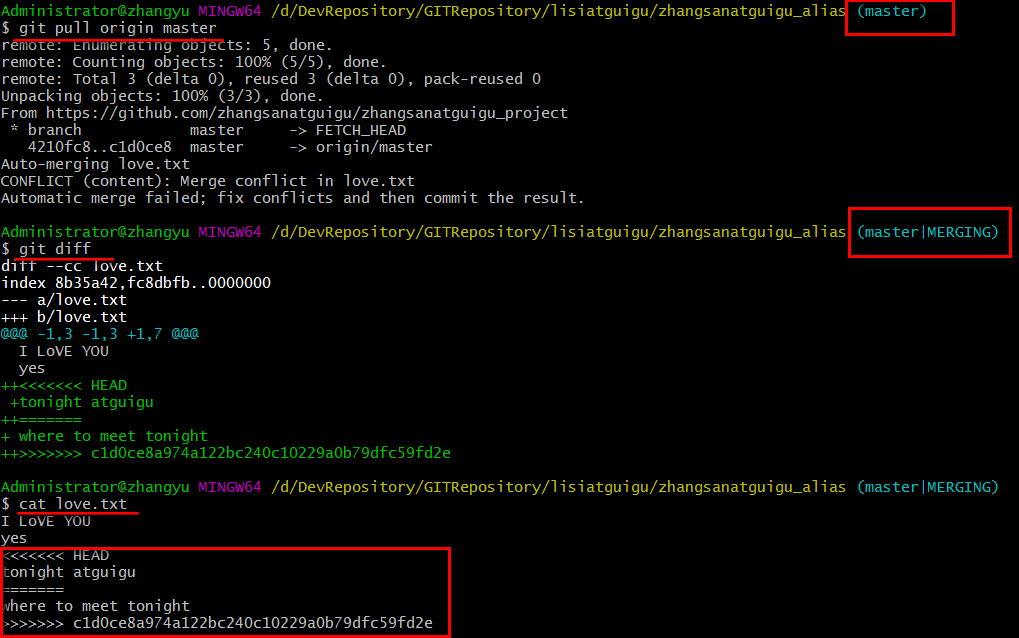
①[zhangsanatguigu]修改并提交到远端，OK.



②[lisiatguigu]修改并提交到远端，NOT OK.



③[lisiatguigu]先pull,合并冲突，再push，OK



计算机生成了可选文字:
Administrator公zhangyu’·’I闪G一／d/DevRepository/GITRepository/lisiatguigu/zhangsanatguigu_alias(masterlMERGING)
5vimlove.txt
Administrator尽zhangyu‘·’工N。；、，:/d/oevRepository/'i丁Repository/lisiatguigu/zhangsanatguigu_alias(masterlM〔R'IN')
Cat1OVe.tXt
LOVLYOU
亡J工
VeS
,here
tomeet'tonTght
tonightatguigu
Administrator公zhangyu'iiN翩64/d/DevRepository拓ITRepos有tory/115有atguigu/zhangsanatguigu_alias(master}M〔RGINC)
5gitaddlove.txt
Administrator公zhangyu翎64
5gitcommit一m”解决冲突”
［币百亏teruj,uctaJ蔫软戴中哭
'd'D二R·pOS,，。ry'",TR·pOS,，。ry',，·，·，g·，g·／·h二g·…，g·，g一，，二匹亘巫亚」
Administrator公zhangyu"
/d/DevRepository/GITRepository/lisiatguigu/zhangsanatguigu_alias
591
t
pushorginmaster
fatal犷
fatal:
orgln
COUld
doesnotappeartobeagitrepository
notreadfromremoterepository.
Pleasemakesureyouhavethecorrectaccessrights
andtherepositoryexists.

## 5.5 Github实战—跨团队协作

### Fork

* **概念:**

现在的情景是，用叉子把别人的东西（copy no cut）叉到你的碗里～

就是把别人的项目clone一份，但是owner变成自己，这样你就可以在遵守Open source license的前提下任意修改这个项目了。

相当于你在原项目的主分支上又建立了一个分支，你可以在该分支上任意修改，如果想将你的修改合并到原项目中时，可以pull request，

这样原项目的作者就可以将你修改的东西合并到原项目的主分支上去，这样你就为开源项目贡献了代码，开源项目就会在大家共同的努力下不断壮大和完善。

* **流程：（zhangsan和lisi是一个团队的人，wangwu是另一个团队的人）**

第三方账号登录（wangwu），搜索某账号找到某项目(zhangsan)，然后点击Fork按钮，这样就将该项目克隆一份到当前账号(wangwu)内，然后进行修改，提交pull request，告诉对方(zhangsan)已经修改了，请求合并；

当前用户登录(zhangsan)后，可以看到小铃铛中的消息提醒，查看到pull request，然后，选择合并或拒绝。

计算机生成了可选文字:
FOFk
渭
问1iequest
、一
I
贾
目口
口口
目口
困

### 演示Fork

①[wangwuatguigu]获取[zhangsanatguigu]的项目URL(可以进行搜索,也可以通过email发送url地址告知)

计算机生成了可选文字:
.),',hU”·〔om"hang'an",g"'g"/zhang'"n",gu'g妇不祖北一1)
翻q
☆晶e:
/PullrequestsIssuesMarketplaceEXplore
户、＋,
〔）
UI刀pto
出戈
习zhangsanatguigu/zhangsanatguigu_project
公W欲ch甲1
★Staro
《》Code①lssueso
九Pullrequestso
画projectso困wiki困security匝Insights
练习项目
05commits
驴1branch
令0releaseS
从2contributors
Branch:master甲Newpullrequest
Cre。，enew",eUp'oad,"es「‘nd,',e[C,oneordown'oad-
Latestcommlto35ocfa26minutesago
解决冲突
26minutesago

②[wangwuatguigu]点击[Fork]，产生分支

计算机生成了可选文字:
00一卜
l[_
9ithub.com/wangwuatguigu/zhangsanatguigu_project
蜘q
SearC"o『卿p'0…了，与
l·几、、、口Pu,'reque,ts
lssuesMark鱿placeEXplore
☆晶e
孕＋甲户
yWangwuatguigu/zhangsanatguigu_project
OW欲ch甲0
★Staro
臂Fork
forkedfromzhangsanatguigu/zhangsanatguigu_prOJect
口
＜》code户pullrequestso
画Projectso
困W1ki
卿security山．Insights今Settings
练习项目
Managetopics
Edit
05commits
F1branch
勺0releaseS
名孟3ContribUtorS
Branch:master,Newpullrequest
Createnew行leUploadfiles
Find石le
Thisbranch15evenwithzhangsanatguigu:master.
只：5iatgu'gu黔中突
国love.tXt
九Pullrequest函compa「e
Latestcommito35ocfa32minutesago
解决冲突
32minutesago
HelppeopleinterestedinthisrepositoryunderstandyourprojectbyaddingaREADME.
Add日README

③[wangwuatguigu]修改项目文件.(可以pull本地后修改,也可以在github上直接进行修改.)

计算机生成了可选文字:
zhangsanatguigu_project/
}OVe.tXt
CanCel
《＞Editfile
公Previewchanges
Spaces尝
2:Soffwrap:
1LOVEYOU
yes
‘、heretomeettonight
留盈台留翅卜－-
1don·tagree一wangwu
1Zj、
恐
Commitchanges
三个人的恋情
三个人的恋情
．心Commitdirectlytothemasterbranch.
＿·扎Createanewbranchforthiscommitandstartapullrequest.Learnmoreaboutpullrequests.
CanCel

④[wangwuatguigu]-> [New pull request]

计算机生成了可选文字:
丫wangwuatguigu/zhangsanatguigu_Project
公Watch,
0★Staro
？「ork
forkedfromzhangsanatguigu/zhangsanatguigu_project
竹Pullrequestso
一l'I一Projectso
lwiki①Security
匝Insights今Settings
练习项目
ManagetoplcS
Ed代
06commits
肛1branch
令0releases
从3contributors
Branch:master,1Newpullrequest
C陀atenewfile
Uploadfiles
Findfile
Thisbranch151commitaheadofzhangsanatguigu:master.
npullrequest函compare
,w。ngwu。tg。．9。针人的旎
Latestcommitsasecfs3minutesago
国Iove.txt
三个人的恋情
3minutesago
HelppeopleinterestedinthisrepositoryunderstandyourprojectbyaddingaREADME.

⑤[wangwuatguigu]->[Create pull request]->[填写pull request信息]

计算机生成了可选文字:
目zhangsanatguigu/zhangsanatguigu_project
OWatch,
★Staro
?rork
＜》Code①lssueso
介Pullrequestso
画Projectso
困wiki叼Security匝Insights
openapullrequest
Createanewpullrequestbycomparingchangesacrosstwobranches.Ifyouneedto,youcanalsocompareacrossforks.
眺1base。epository::h。ngs。natguig。／zh。ngsanatg一base:m。ste：甲1.1headreposit。「y:wangwuatguigu,zhang臼natgu一compare:master,
了Abletomerge.Thesebranchescanbeautomaticallymerged.
努
三个人的恋清
Write
PFeVieW
AABi
"<＞龟
：三注乍＠日气，
Attachfilesbydragging&dropping.selectingorpastlngthem.
‘硬1AI}oweditsfrommaintainers.Learnmore
Createpullrequest
咬＞1Commit
函1filechanged
口0commitcomments
从1contributor
印Commitsonoct25,2019
。，w。ngwuatgu.gu
三个人的恋情
Verifi曰5己SeCfS

⑥分支已提交到主分支

计算机生成了可选文字:
目zhangsanatguigu/zhangsanatguigu_project
OWatch,
奋Staro
?Fo改1
《＞Code①Issueso
九Pullrequestsl
囚ProjectSO
匝IWiki嘎二Security匝Insights
三个人的恋情＃1
Edit
匝画卜。ngwu。，gu'guw,n,',0me『ge,'omm',,n,0·h二g·…，g·，g一，二，『om…“二，g·，‘一，二，
夕Conversationo
峨＞CommitSI
县Checkso
函Fileschanged'
+1一0.
资
wangwuatguigucommented5minutesago
＋⑧
ReVieweFS
NOFeVIeWS
三个人的恋情（写清楚「ork提交的原因
。勺三个人的恋情
Assignees
Nooneassigned
V饱rified
5己seCfS
L日bels
Addmorecommitsbypushingtothemasterbranchonwangwuatguigu/zhangsanatguigu_project.
Noneyet
卜
Thisbranchhasnoconflidswiththebasebranch
石袱尸饰O亏e、气l．不毛卜、．，、rr了t百飞沱艺国抢
15repo引ToryC日nmeFgepU!1reqUeSTS·
projects
Noney鱿
卜

⑦[zhangsanatguigu]和[lisiatguigu]都可以看到[wangwuatguigu]提交的项目内容.

计算机生成了可选文字:
目zhangsanatguigu/zhangsanatguigu_Project
公Watch甲
0Yro比
<>Code
!SSUeSO
九Pullrequests
l'l一Projectso
W1ki.)Security
互Insights
三个人的恋情＃1
Edit
巫回wongw。。，g。，guw"n,',om，『g,,'omm','n,0·h二g·…，：·，g一，二，『om…g…，g·，g一，二良
夕Conversationo
县Checkso
+1一0.
Changesfromallcommits,FIIefilter．二，Jumpto一O,
0/1filesviewed
Revi眯cha叼eS,
气尹1.
工ove.txt良
曰VIewed
由
卵一2,3+2,4卵1LovEYoU
yes
',heretomeettonight
tonightatguigu

⑧[zhangsanatguigu]合并项目,点击[Merge pull request]->点击[Confirm merge]

计算机生成了可选文字:
目zhangsanatguigu/zhangsanatguigu_project
()Unwatch,1★Staro
甲Forkl
《》Code
①lssueso
竹pullrequestsl
画Projectso
~{Wiki
霉Security匹Insights
今Settings
三个人的恋情＃1
应画wongwu。，gu'guWon,s,0me『g,,Comm','n,o:h二95…，g·，：一，二from…。二，‘·，“一，二，
夕Conversationo
咬＞CommitSI
县Checkso
函Fileschanged'
ommented14minutesago
First一timeCOntribUtor
＋⑧
ReViewerS
Edit
+1一0.
今
Suggestions
只‘,,,a,gu'g"
肋quest
。努三了队。恋情
Ventled
5己se仁＋5
AssigneeS
Noone一assignyourself
今
Addmorecommitsbypu、hingtothemasterbranchonwangwuatguigu/zhangsanatguigu_project.
比bels
今
卜
Noney鱿
函continuousintegrationhasnotbeensetup
Severalappsareavailabletoautomaticallycatchbugsandenforcestyle.
projects
今
Noneyet
.
Thisbranchhasnoconflictswiththebasebranch
Mergingcanbeperformedautomatically.
Mi!estone
今
NOmilestone
YOucanalsoopenthlsinGitHubDesktopor训ewcommandIineInstructions.
Notifi(ations
CUStomiZe
卜

⑨[zhangsanatguigu]查看项目代码，合并完成，OK

计算机生成了可选文字:
目zhangsanatguigu/zhangsanatguigu_project
()Unwatch甲
*sta。。一？「。比
《》Code①Issueso
负Pullrequestsl
画Projectso
困wiki吻Security匝Insights今Settings
三个人的恋情＃1
匡Me。川zhongson。，gu'gumerged，〔omm','n,o·h二g·…，：·，一，二，『om…：…，g·，g一，二，,m'n",,"90
夕Conversationo
峨＞CommitSI
县Checkso
函Fileschanged
+1一0.
降
wangwuatguigucommented17minutesago
contributor＋⑧
ReVieweFS
NOFeVIeWS
三个人的恋情（写清楚「ork提交的原因
。，三个人的恋清
AssigneeS
Noon卜assignyou巧elf
今
VeFifi州saseCfS
L日be!S
今
J口卜，
月曰1衬zn""gsa""tg"'g"me『g"aComm'tb47c6cd'"TO
zhangsanatguigu：爪aster1minuteago
ReVert
Noney鱿

计算机生成了可选文字:
目zhangsanatguigu/zhangsanatguigu_Project
()Unwatch,
★Staro
VFork
《》Code①lssueso脚Pullrequestso画Projectso
Wiki嘎二‘Security业Insights今Settings
B「an(h:maste「
,zhangsanatguigu一roject/love.txt
Findfile
Copypath
,wangwuatg。．9。三个人的恋情
4'on，『，bu,ors只：言，
sasecfS32minutesago
6line:(5sloc)75Bytes
RawBlameHistory
甲，画
1LOVE丫OU
yes
wheretomeettonight
tonightatguigu
1don·tagree一wangwu

## 5.6 解决重复输入密码问题

### 两种模式：https VS ssh

ssh模式比https模式的一个重要好处就是，每次push，pull，fetch等操作时不用重复填写用户名密码。

前提是你必须是这个项目的拥有者或合作者，且配好了ssh key

### 如何配置ssh key

①步骤1：检查电脑家目录下是否已经生成了SSH key，如果有.ssh目录删除即可

②步骤2：创建SSH Key : ssh-keygen -t rsa -C lisiatguigu

成功的话(执行命令不需要输入任何内容)，会在~/下生成.ssh文件夹，查看生成两个文件

id\_rsa

id\_rsa.pub

打开id\_rsa.pub，复制里面的内容，到github上

③步骤3：登录[lisiatguigu]账户

打开用户设置

点击[SSH and GPG keys] 填写Title和Key，点击[Add SSH key]

计算机生成了可选文字:
Personalsettings
SSHkeys/Addnew
Profile
ACCOUnt
Security
Key
EmailS
NotifiCationS
Billing
8)OCkedUSe「S
SSh一rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQC2YInSuxGEteXQ09pE0IV1IvWKeFZguX1UBoozm2dbldJNXyL0YCrAkq4TZad
E45glrCYSCGvrbhoZX8dXACgRkQCKiw+ZXydN7HDWf5)+/gdUpAIVILVCYIJQTZdKt60MRgl以vgswr4lhUwGosrvKBR
7DdNvqjoSNQS作h4a6flvST+h3QIX427hEOQnDZElyo375psuAGx+DfJKLtaNFlldrDnJmBRGGEXieofA49RKIXpx08T0hI
jNwpadyYBYStZRRgGSUe1FD以6juVOT6gDyVISI以1Gn0ryCJgbVBbkwnTgcVZ0aWWS3WlBoqsd1+entgZe+n4aCMTZ
37H7AiNWb6UsmtasgeLMSTADq1mg2RdDGGQD63cxKCyCmfIpLkim500x/RZCVIKX/rHmxo/qHO/Xwl印iq7zZ+MZsR
MEPx0NTC1TwlHjFxCsHGbqBeRjSLWYARcvXqiK2FIXT3Kdu/DsZCQR/7XEPUWHHPdjmCzZll3igSGGisaodU=
11引atguigu
Repositories
organizations
卜ddssok,1
Savedreplies
ApplicationS

### 重新测试：提交不再需要输入密码

①[lisiatguigu]复制SSH的url连接

|  |  |  |
| --- | --- | --- |
|  | >> |  |

②[lisiatguigu]重新建立github连接,如果之前配置过则删除，重新配置

命令：

git remote add originssh git@github.com:zhangsanatguigu/zhangsanatguigu\_project.git

git push originssh master

修改文件，重新提交,不需要再输入用户名和密码了

### 注意问题:

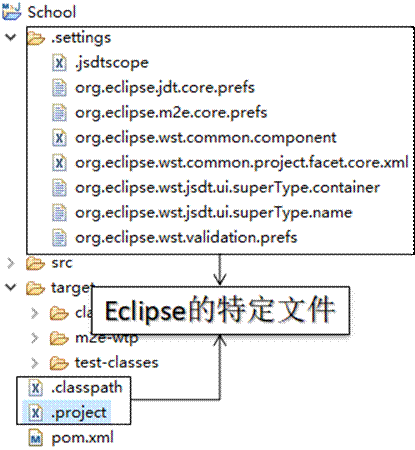
一个账户只能设置一个默认密码.

如果有多个账号存在的话，只能选择设置其中某一个

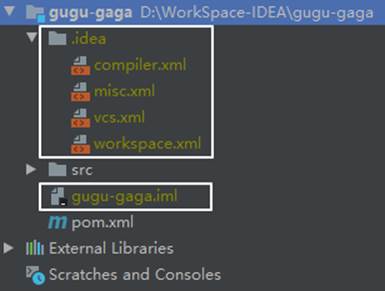
## 5.7 配置Git忽略文件

### 哪些文件要忽略？

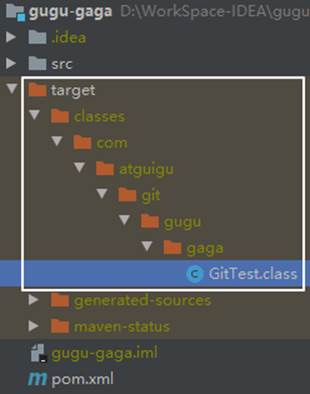
#### Eclipse特定文件



#### IDEA特定文件



#### Maven工程的target目录



### 为什么要忽略它们？

与项目的实际功能无关，不参与服务器上部署运行。把它们忽略掉能够屏蔽IDE工具之间的差异。

### 怎么忽略？

#### 创建忽略规则文件（git.ignore）

这个文件的存放位置原则上在哪里都可以，为了便于让~/.gitconfig文件引用，建议也放在用户家目录下

|  |
| --- |
| # Compiled class file  \*.class    # Log file  \*.log    # BlueJ files  \*.ctxt    # Mobile Tools for Java (J2ME)  .mtj.tmp/    # Package Files #  \*.jar  \*.war  \*.nar  \*.ear  \*.zip  \*.tar.gz  \*.rar    # virtual machine crash logs, see <http://www.java.com/en/download/help/error_hotspot.xml>  hs\_err\_pid\*    .classpath  .project  .settings  target  .idea  \*.iml |

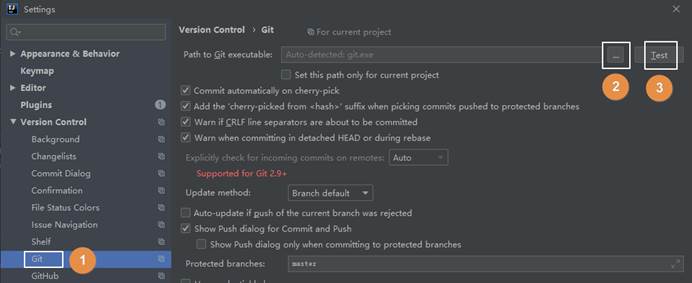
#### 在.gitconfig文件中引用忽略规则文件

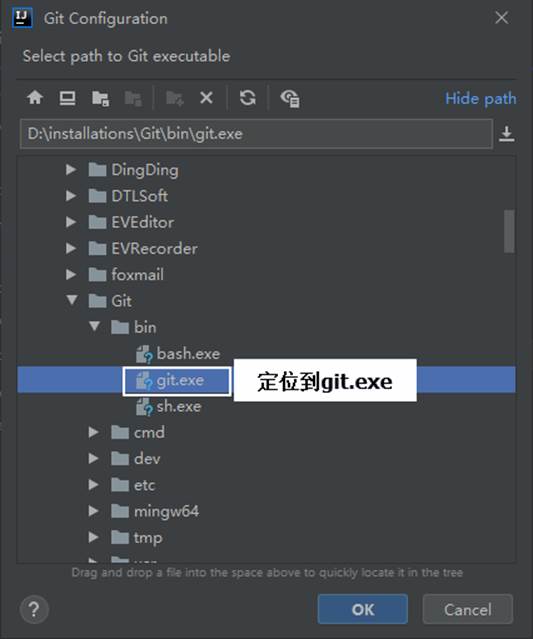
|  |
| --- |
| [user]  name = peter  email = peter@atguigu.com  [core]  excludesfile = C:/Users/Lenovo/git.ignore |

注意：这里要使用“正斜线（/）”，不要使用“反斜线（\）”

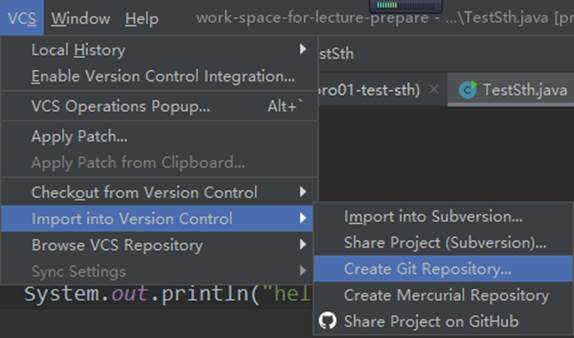
# 第六章 IDEA集成Git

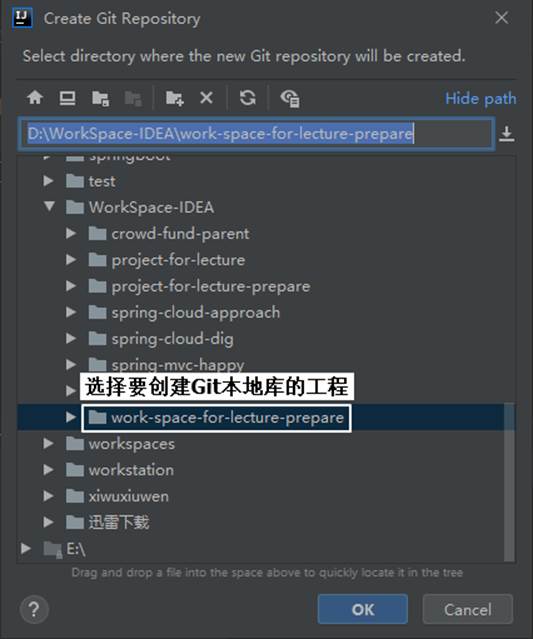
## 6.1 定位Git程序



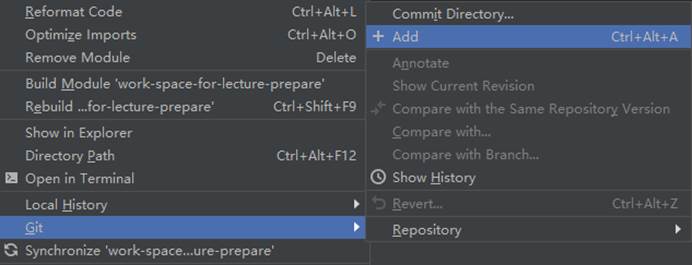


## 6.2 初始化本地库

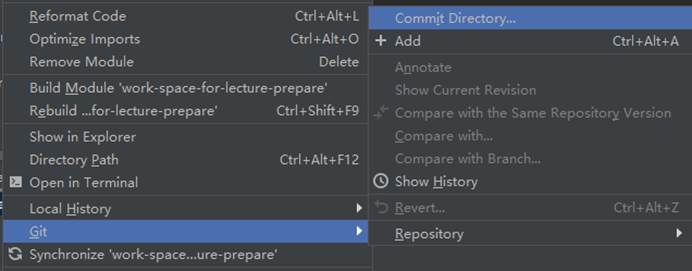


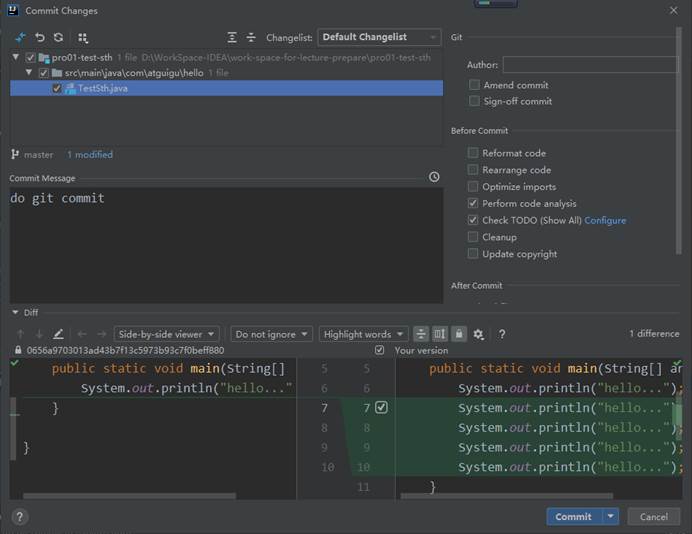


## 6.3 添加到暂存区

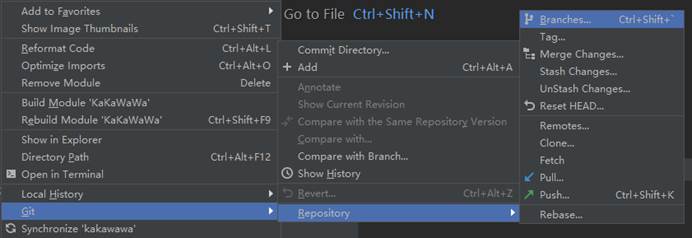


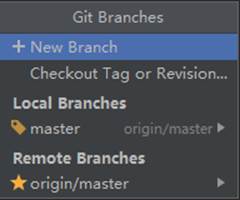
## 6.4 提交到本地库

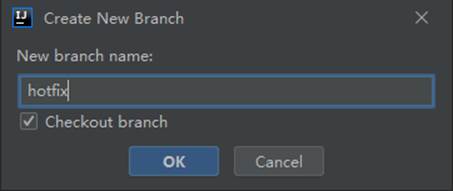




## 6.5 创建分支

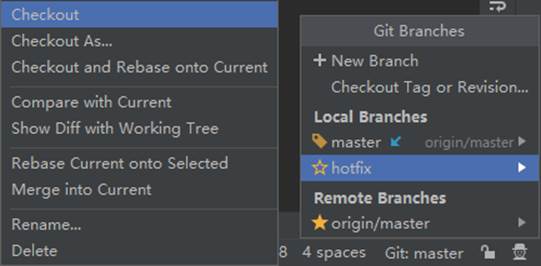




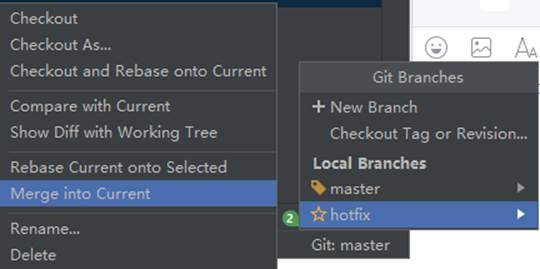


## 6.6 切换分支

在IDEA窗口的右下角



## 6.7 合并分支

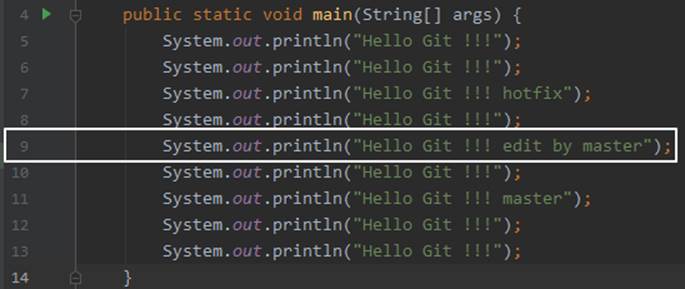


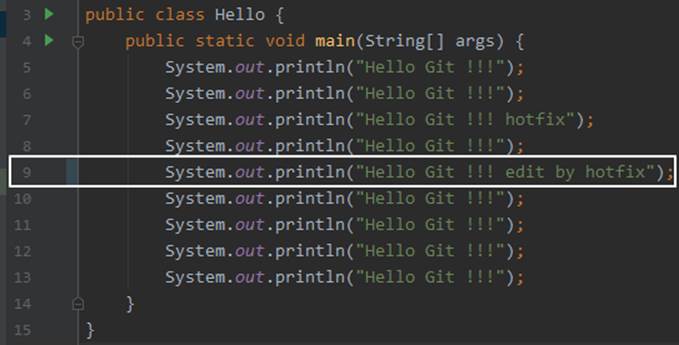
## 6.8 解决冲突

在和远程库交互过程中也有可能发生冲突，因为解决办法一样，所以这里按照分支操作过程中产生的冲突来演示。

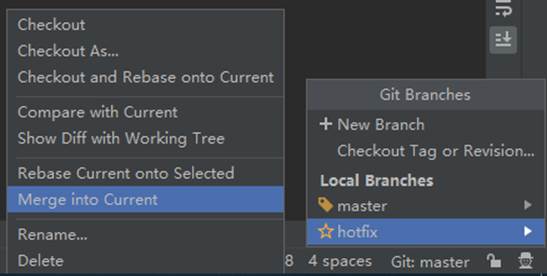
### 冲突的产生

仍然是同一个文件在同一个位置有不同内容。

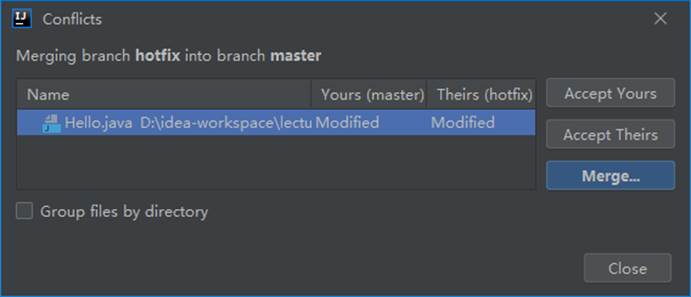




### 合并分支

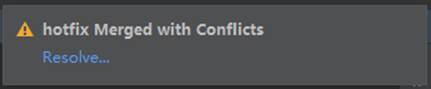


### 看到冲突提示

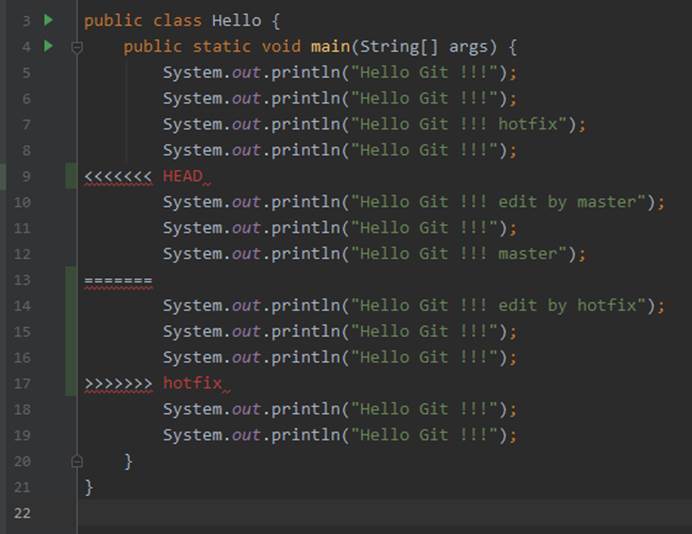


这里可以直接点“Accept Yours”或者“Accept Theirs”来解决冲突。这代表完全使用我们的内容或完全使用他们的内容。

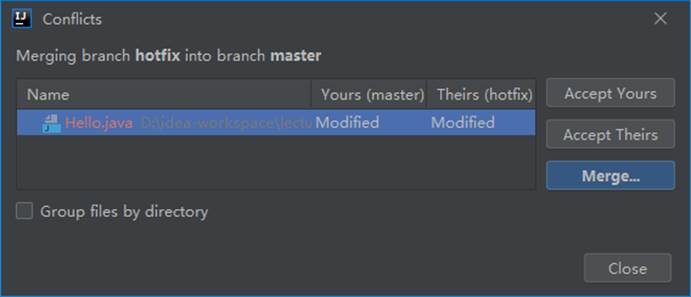
如果我们想要详细修改冲突内容则点击Close。此时IDEA会提示Resolve



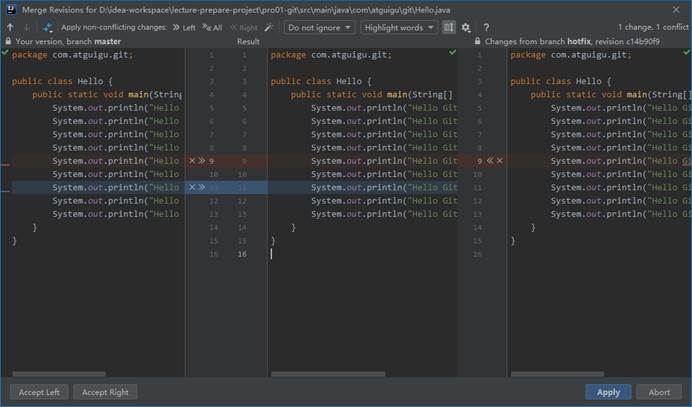
### 冲突的表现



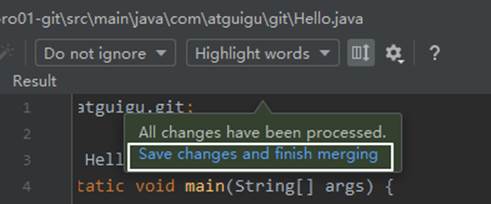
### 点击Resolve



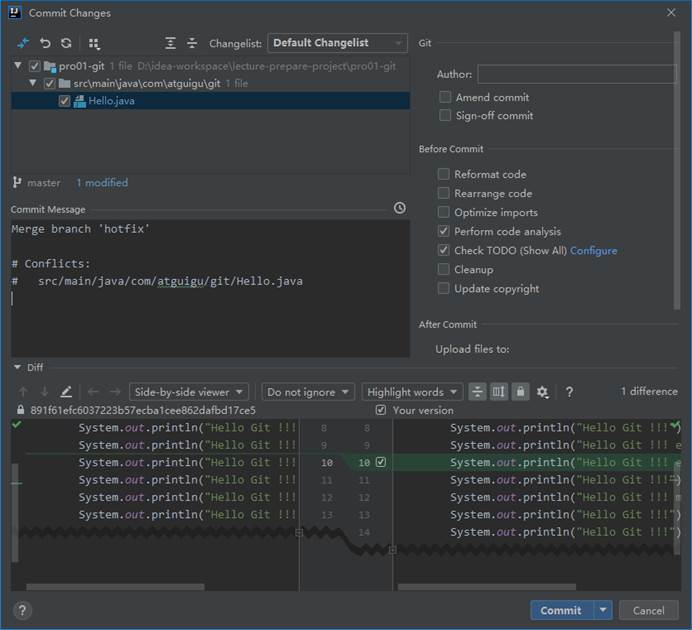
点击Merge...



根据需要调整完成后点击“Save changes and finish merging”

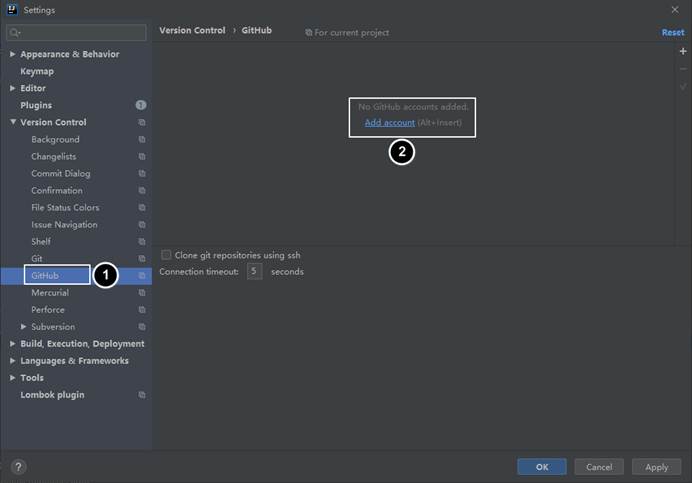


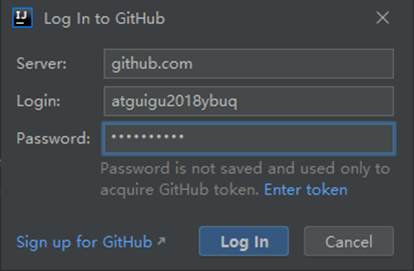
然后提交到本地库就好啦！



# 第七章 在IDEA中连接GitHub

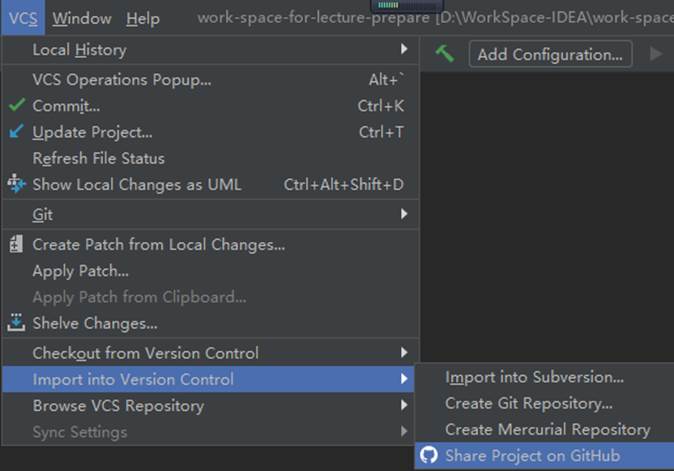
## 7.1 保存GitHub账号

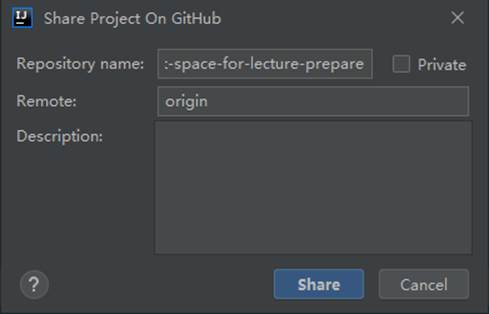


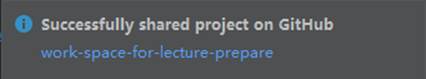


 https://blog.csdn.net/qq\_40820382/article/details/109037084

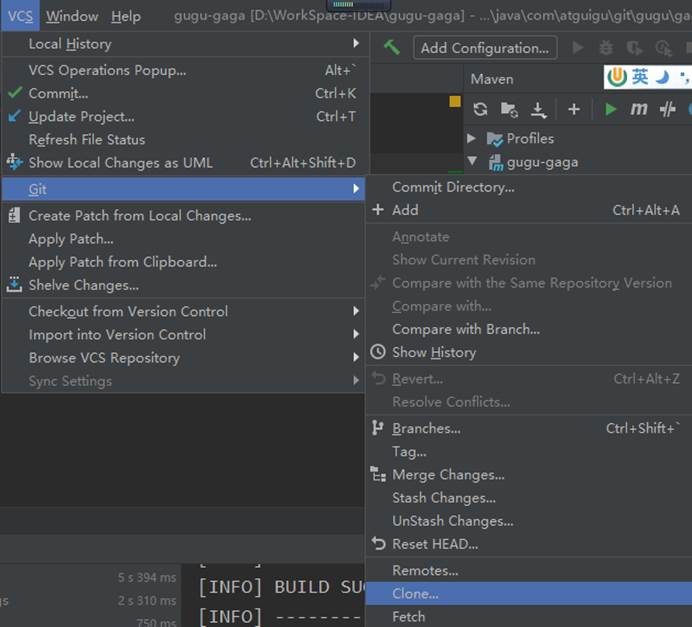
## 7.2分享工程到GitHub

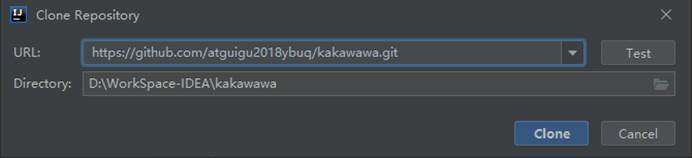


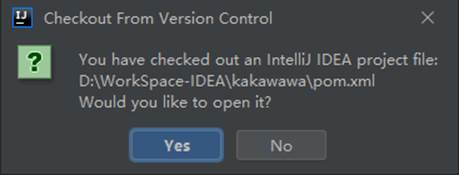


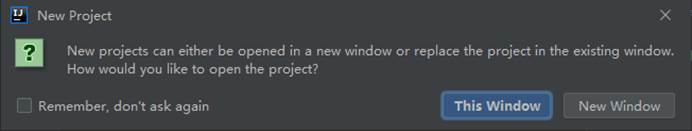


## 7.3 Clone

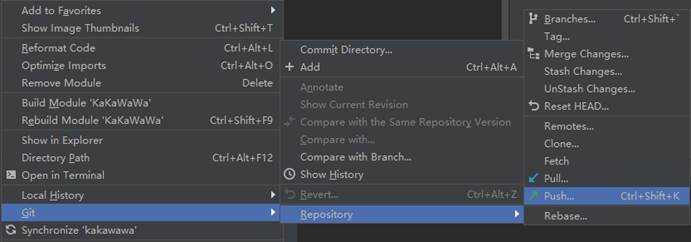




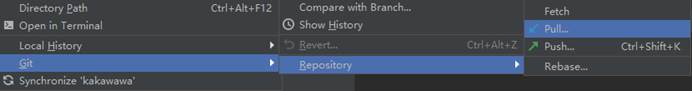


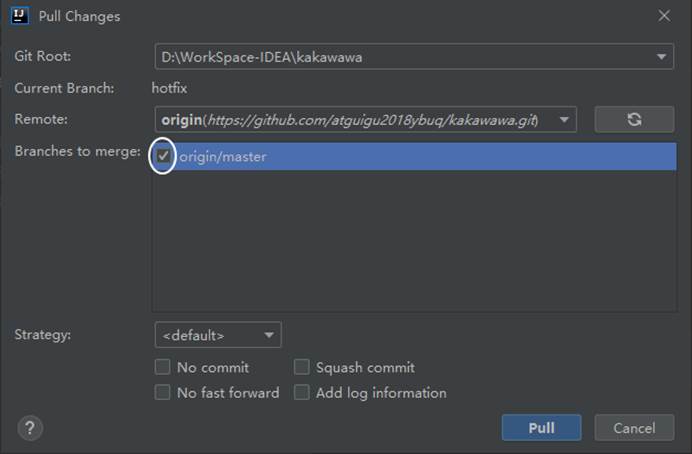


## 7.4 Push



## 7.5 Pull





# 第八章 Git工作流

## 8.1 Git工作流

简单来说就是，一个项目的成员们在工作中统一使用Git的工作方式。

Gitflow工作流通过为功能开发、发布准备和维护设立了独立的分支，让发布迭代过程更流畅。严格的分支模型也为大型项目提供了一些非常必要的结构。

* **简单的分支**

计算机生成了可选文字:
罕份＊.-
MSSt6r
令．+.---－争．

* **复杂分支案例演示：**

计算机生成了可选文字:
Git工作流
.'r
训。O
mast.r.
V,
二
月口口价口口口口口口口．
~...．卜
Bug!
呢．0
二
甸匀．0
PaSS!
二公．汗
尹、
hottiX
。．
、飞．
r目．台56，友0
d.v.lop
佳atur._goldstyl.
/
feature_game
....．卜．...．卜……卜．

## 8.2 分支种类

* 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

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。

# 第九章 Gitlab私服

注意：要使用CentOS7版本安装，CentOS6版本不行。

## 9.1 官网地址

首页：https://about.gitlab.com/

安装说明：https://about.gitlab.com/installation/

## 9.2 提前下载所需rpm

yum安装gitlab-ee(或ce)时，需要联网下载几百M的安装文件，非常耗时，所以应提前把所需RPM包下载并安装好。

下载地址是：https://packages.gitlab.com/gitlab/gitlab-ce/packages/el/7/gitlab-ce-10.8.2-ce.0.el7.x86\_64.rpm

下载好后上传到Linux系统，习惯上还是放在/opt目录下

## 9.3 安装步骤

|  |
| --- |
| sudo rpm -ivh /opt/gitlab-ce-10.8.2-ce.0.el7.x86\_64.rpm  sudo yum install -y curl policycoreutils-python openssh-server cronie  #sudo yum install lokkit  sudo lokkit -s http -s ssh  sudo yum install postfix  #yum install keyutils-libs-devel libcom\_err-devel libselinux-devel  libverto-devel -y  #rpm -ivh mysql-community-libs-compat-5.7.28-1.el7.x86\_64.rpm  sudo systemctl enable postfix.service  sudo chkconfig postfix on  curl <https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.rpm.sh> | sudo bash  sudo EXTERNAL\_URL="http://gitlab.example.com" yum -y install gitlab-ce |

当前步骤完成后重启。

## 9.4 Gitlab服务操作

* 初始化配置

gitlab-ctl reconfigure

* 启动Gitlab服务

gitlab-ctl start

* 停止Gitlab服务

gitlab-ctl stop

## 9.5 浏览器访问

访问Linux服务器IP地址即可，如果想访问EXTERNAL\_URL指定的域名还需要配置域名服务器或本地hosts文件。

初次登录时需要为gitlab的root用户设置密码。

