

# Git&GitHub

(分布式版本控制工具及代码托管中心)

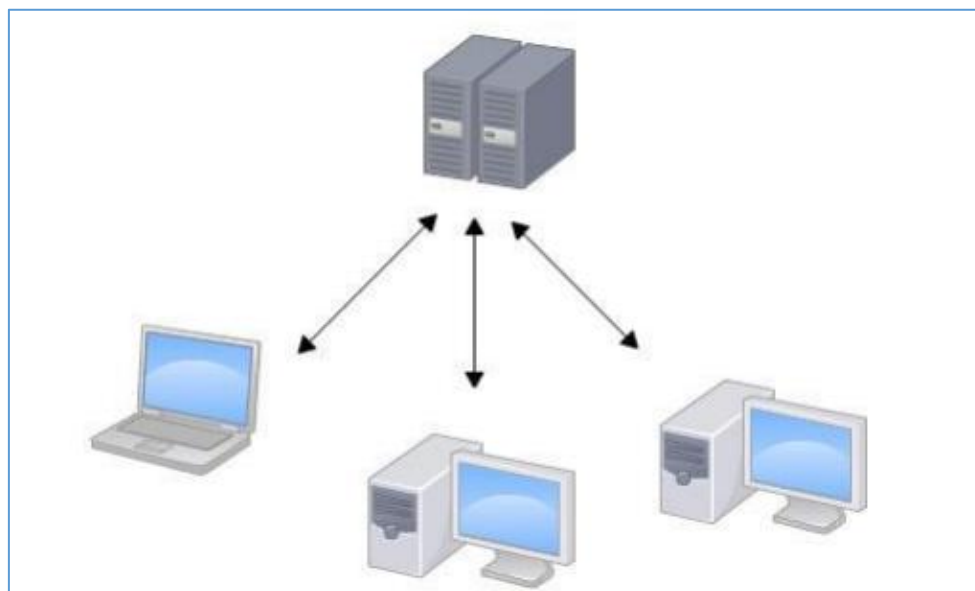
---



- Git是一种分布式版本控制工具
- 主要功能：
  - 团队协作开发
  - 版本管理
  - 权限控制
  - 分支管理
  - 查看历史记录

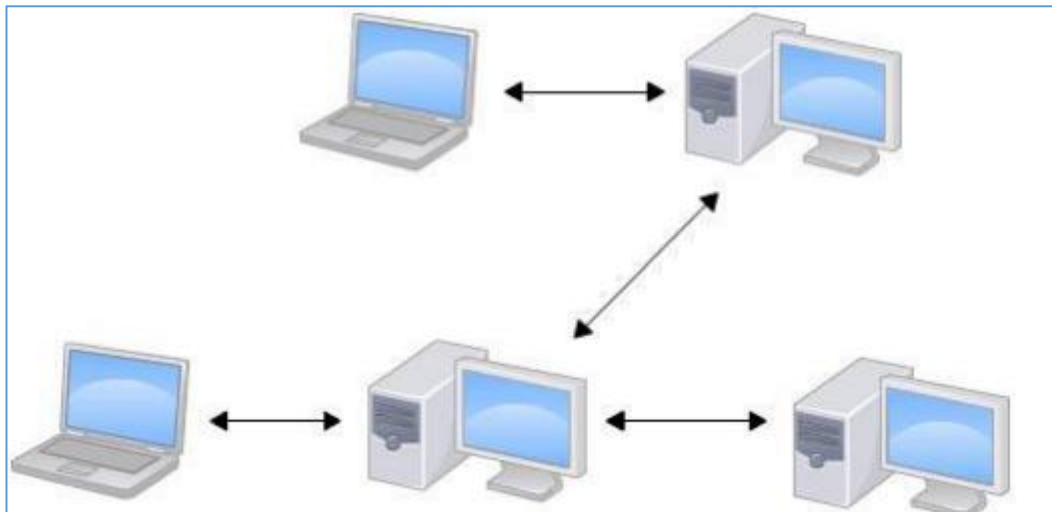
# Git简介——与SVN区别

- SVN是集中式版本控制
  - 所有人的代码都提交到同一个服务器



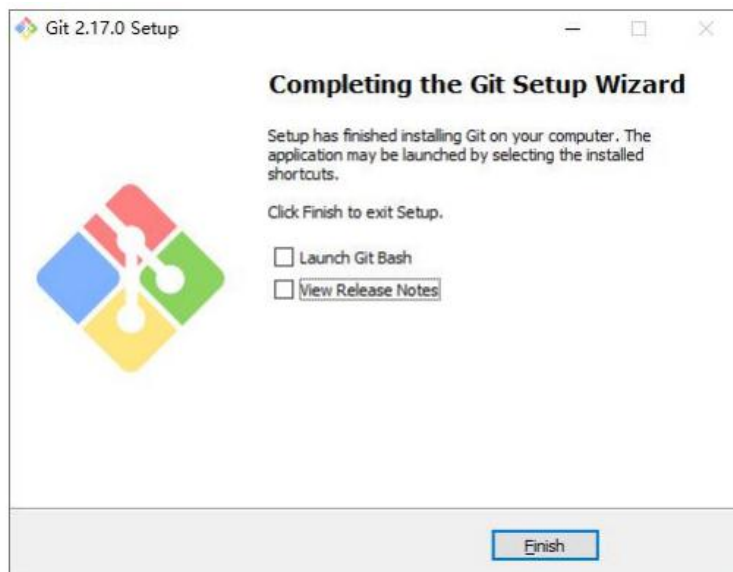
- 优点：操作方便，本地只需安装客户端
- 缺点：一旦服务器损坏则代码全部丢失

- Git是一种分布式版本控制工具
  - 每个人都在本地都有一个本地库用于存放代码

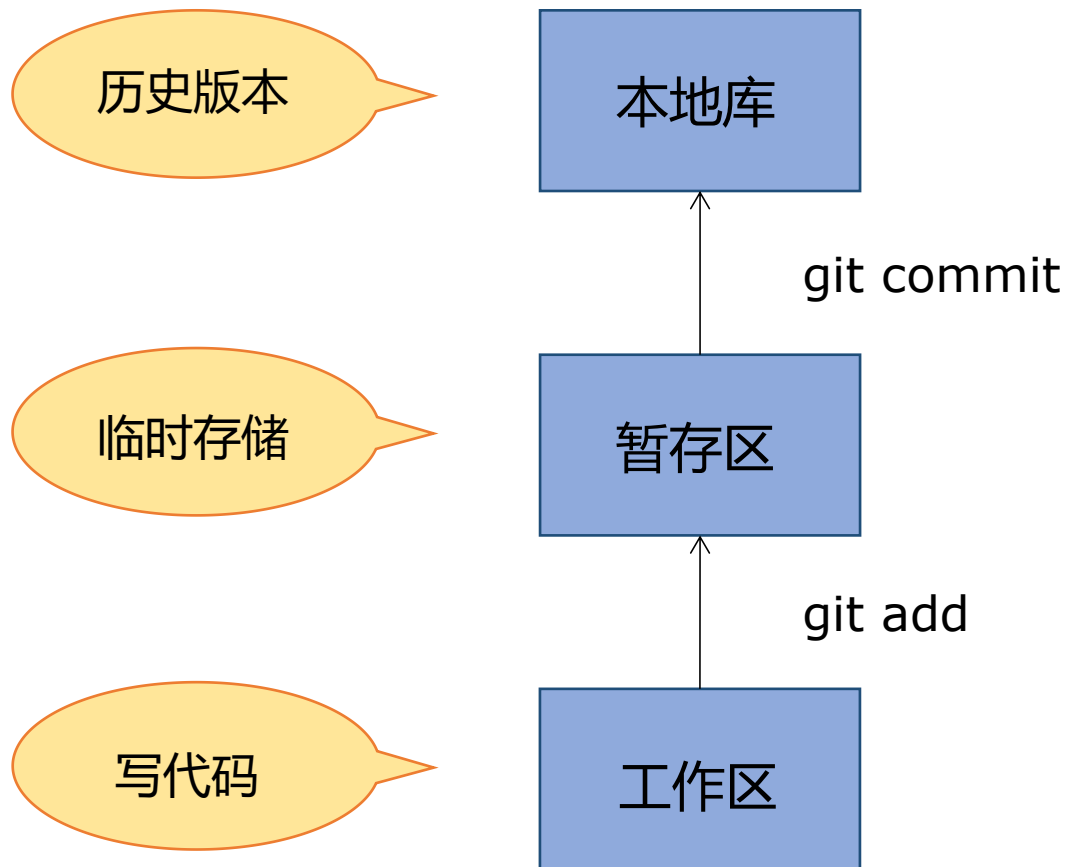


- 优点：大部分的操作都可以在本地完成，无需联网，可以保证完整性  
分支操作非常快捷流畅  
与Linux命令操作全面兼容
- 缺点：需要在每个人电脑创建本地库

- 官网下载最新版本 <https://git-scm.com>
- 选择非中文目录，按照默认选项安装即可



- 本地的Git结构：

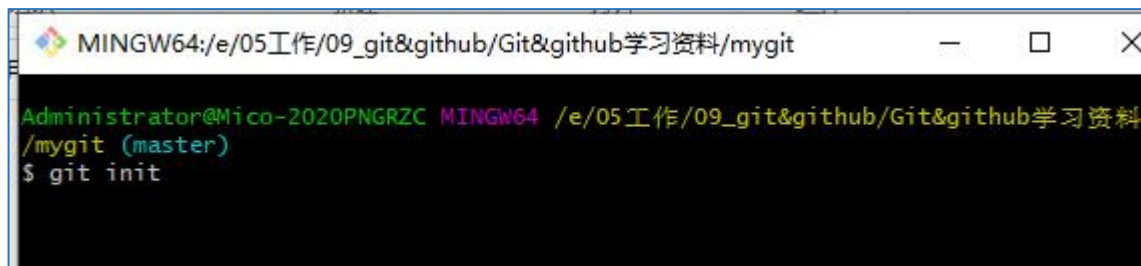


- 在电脑任意位置右键：
- Git GUI Here：
  - 采用图形界面的方式进行Git操作
- Git Bash Here：
  - 采用命令行的方式进行Git操作
- 因为Git与Linux命令全面兼容，所以命令方式操作更加强大，并且提示信息非常详细但是要求一定的Linux基础。



# Git创建本地库

- 命令方式
  - 进入特定文件夹，右键选择 Git Bash Here，打开命令行
  - 输入git init，那么当前文件夹就被指定为本地库了
  - 并且该文件夹是作为整个本地库的master分支，也就是主分支

A screenshot of a Windows command prompt window titled "MINGW64:/e/05工作/09\_git&github/Git&github学习资料/mygit". The prompt shows the user is "Administrator@Mico-2020PNRZC" and the current directory is "/e/05工作/09\_git&github/Git&github学习资料/mygit (master)". The command "\$ git init" has been entered and executed, resulting in a new line prompt "\$".

```
MINGW64:/e/05工作/09_git&github/Git&github学习资料/mygit
Administrator@Mico-2020PNRZC MINGW64 /e/05工作/09_git&github/Git&github学习资料
/mygit (master)
$ git init
```

注意：该目录下有一个.git的隐藏文件，存放本地库香瓜子目录和文件，不可修改或者删除



# 创建用户

- 创建开发人员账号
  - `git config user.name lxr`
  - `git config user.email lxr@qq.com`



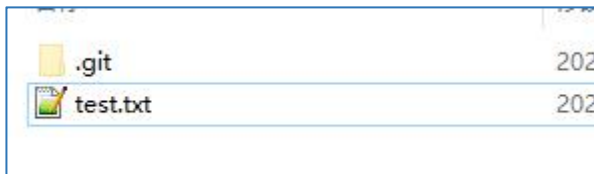
```
Administrator@Mico-2020PNGRZC MINGW64 /e/05工作/09_git&github/Git&github学习资料/mygit
/mygit (master)
$ git config user.name lxr

Administrator@Mico-2020PNGRZC MINGW64 /e/05工作/09_git&github/Git&github学习资料
/mygit (master)
$ git config user.email lxr@qq.com

Administrator@Mico-2020PNGRZC MINGW64 /e/05工作/09_git&github/Git&github学习资料
/mygit (master)
$ |
```

# 创建文件

- 在本地库下创建测试文件test.txt



- 在bash下查看当前状态 git status

```
Administrator@Mico-2020PNGRZC MINGW64 /e/05工作/09_git&github/Git&github学习资料/mygit
$ git status
On branch master

No commits yet

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   test.txt
```

- 提示：在主分支上，本地库没有提交的内容，暂存区没有要提交的内容  
但是在工作区有一个文件可以使用git add添加到暂存区

- 将文件添加到暂存区 `git add [文件名]`

```
Administrator@Mico-2020PNGRZC MINGW64 /e/05工作/  
/mygit (master)  
$ git add test.txt
```

- 然后将文件提交到本地库 `git commit -m "注释" [文件名]`

```
Administrator@Mico-2020PNGRZC MINGW64 /e/05工作/  
/mygit (master)  
$ git commit -m "first commit" test.txt  
[master (root-commit) df29842] first commit  
1 file changed, 1 insertion(+)  
create mode 100644 test.txt  
  
Administrator@Mico-2020PNGRZC MINGW64 /e/05工作/  
/mygit (master)  
$ git status  
On branch master  
nothing to commit, working tree clean
```

- 提交完成再使用`git status`查看状态

## 其他命令操作

- git log 查看历史记录

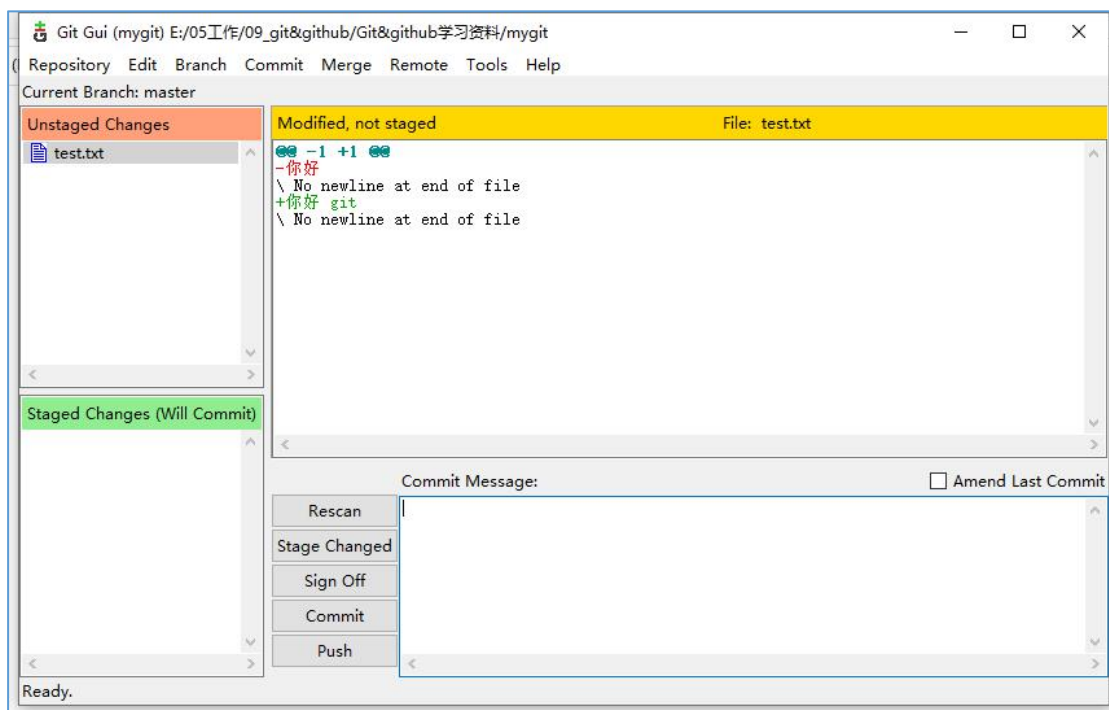
```
Administrator@Mico-2020PNGRZC MINGW64 /e/05工作/09_git&github/Gi
/mygit (master)
$ git log
commit df298427c89f3f2c82dabfb22c1ce7f49c1b2aca (HEAD -> master)
Author: lxr <lxr@qq.com>
Date:   Wed Feb 19 12:34:38 2020 +0800

    first commit
```

- git diff [文件名] 比较工作区与本地库的文件差异

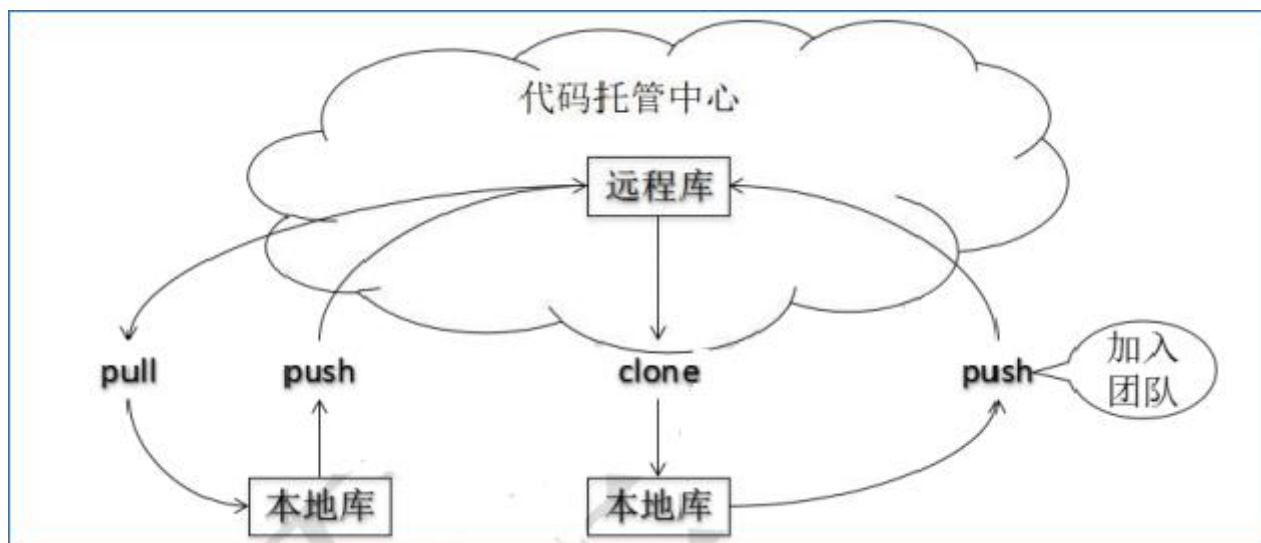
```
Administrator@Mico-2020PNGRZC MINGW64 /e/05工作/09_git
/mygit (master)
$ git diff test.txt
diff --git a/test.txt b/test.txt
index f7b8dbe..ce5d970 100644
--- a/test.txt
+++ b/test.txt
@@ -1,1 @@
-<C4><E3><BA><C3>
\ No newline at end of file
+<C4><E3><BA><C3> git
\ No newline at end of file
```

- Git本身提供的GUI（图形界面工具）也可以完成这些工作，但是提示信息相对来说没有那么详细，推荐使用命令方式。



- 也可以使用第三方的图形工具，比如SourceTree，操作更加方便。

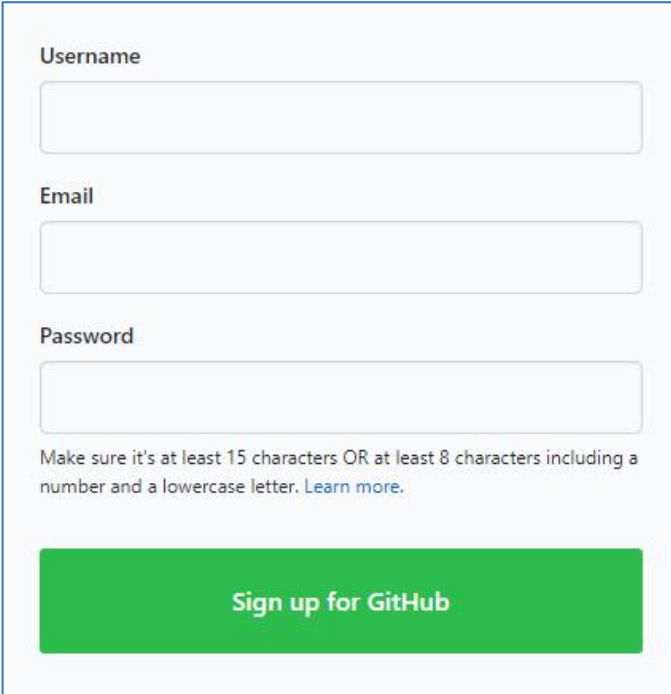
- 只有本地库并不能很好进行团队协作开发，所以还需要一个远程库
- 所有本地库都可以和远程库进行交互



- Git创建远程库方式
  - 公司局域网：GitLab，GitLab需要自行搭建
  - 外网：GitHub、码云
  - GitHub和码云是外网的一个代码托管中心，我们可以直接在这里创建属于自己的远程库。
  - 首先需要注册账号，然后创建远程库。
- 远程库与本地库的交互
  - push：将本地库代码推到远程库
  - pull：将远程库代码拉取到本地库
  - clone：将远程库代码复制到一个新的本地库

# 创建GitHub账号

- 登录GitHub <https://github.com>
- 输入信息进行注册操作
  - 邮箱必须真实有效，需要接收邮件进行账号验证
  - 最好不要是163邮箱，有可能无法正常接收邮件

A screenshot of the GitHub registration form. It features three input fields for 'Username', 'Email', and 'Password'. Below the password field is a note: 'Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)'. At the bottom is a green button labeled 'Sign up for GitHub'.

Username

Email

Password

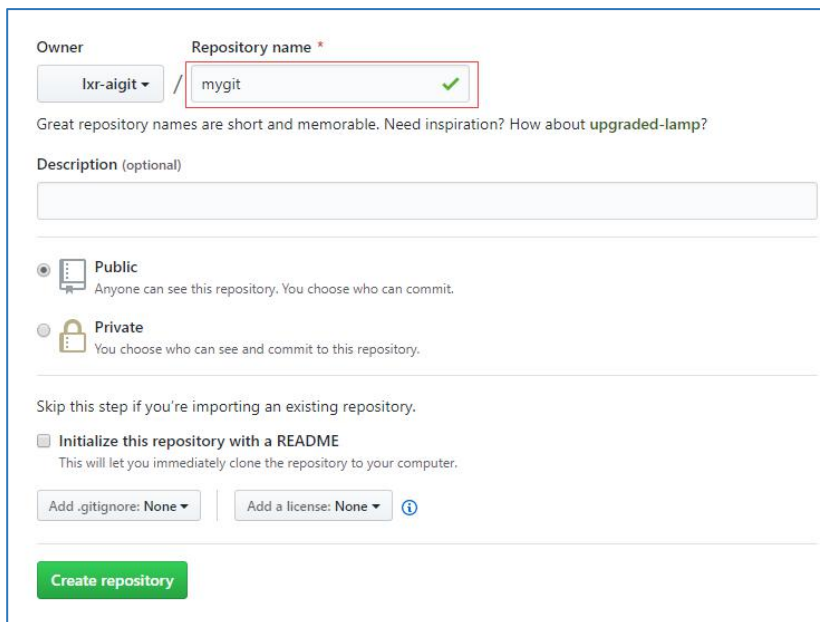
Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Sign up for GitHub



# 创建远程库

- 填写远程库名称，创建远程库



Owner: lxr-aigit / Repository name: mygit ✓

Great repository names are short and memorable. Need inspiration? How about **upgraded-lamp**?

Description (optional)

☒ **Public**  
Anyone can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

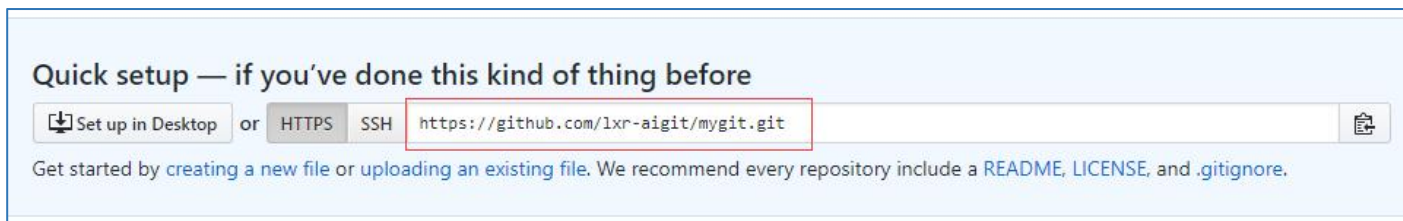
Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer.



Add .gitignore: None | Add a license: None ⓘ

**Create repository**

- 远程库地址



**Quick setup — if you've done this kind of thing before**

 Set up in Desktop or **HTTPS** **SSH** `https://github.com/lxr-aigit/mygit.git` 

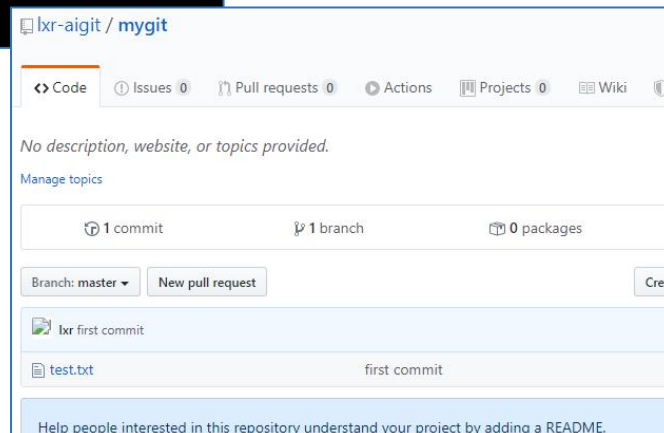
Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

# 将本地库文件推送到远程库

- 打开bash，输入命令 `git push [远程库地址] [分支名]`
- 这一步需要输入GitHub账号

```
Administrator@Mico-2020PNGRZC MINGW64 /e/05工作/09_git&github/G
/mygit (master)
$ git push https://github.com/lxr-aigit/mygit.git master
Logon failed, use ctrl+c to cancel basic credential prompt.
error: unable to read askpass response from 'C:/Program Files/G
c/git-core/git-gui--askpass'
Username for 'https://github.com': lxr-aigit
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 203 bytes | 101.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/lxr-aigit/mygit.git
 * [new branch]      master -> master
```

- 推送成功后可以在 远程库看到被推送的内容



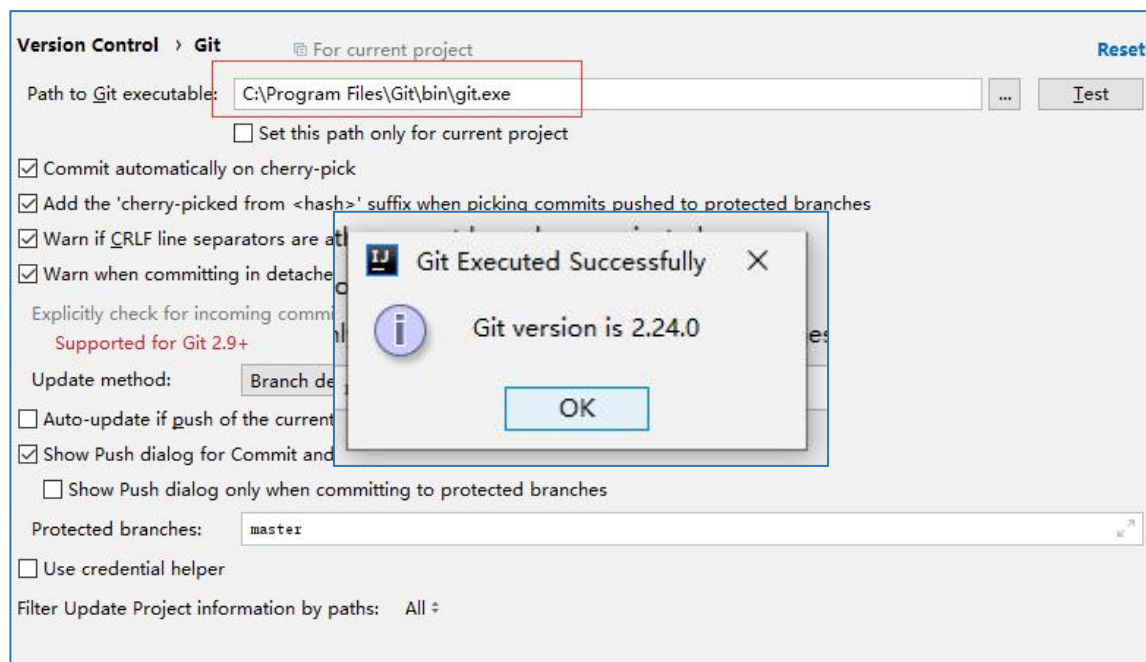
# 将远程库克隆到本地库

- 推送成功后，其他人可以将远程库内容克隆到本地
- `git clone [远程地址]`

```
$ git clone https://github.com/atguigu2018ybuq/huashan.git
Cloning into 'huashan'...
remote: Counting objects: 10, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 10 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (10/10), done.
```

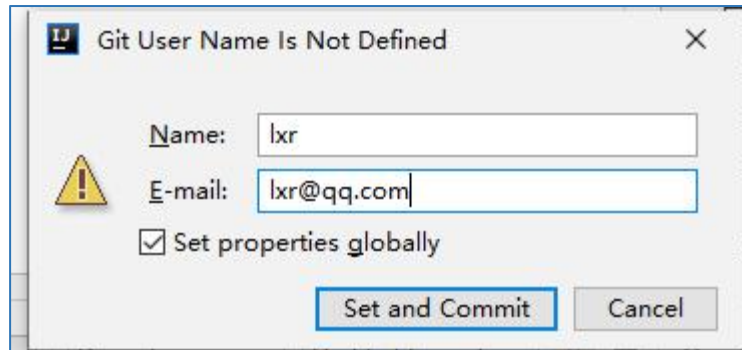
- 通过这样的方式，可以团队协作开发。

- 配置Git
  - 选择file-Settings-Version Control-Git
  - 选择到git.exe目录，点击Test，弹出成功对话框即为配置成功。



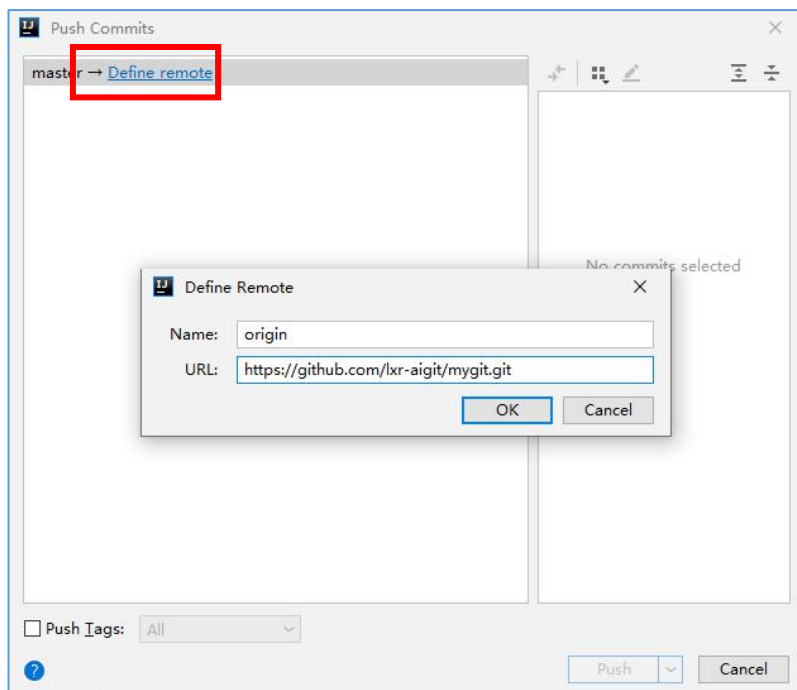
# 创建项目并提交到本地库

- 创建idea项目，选择VCS--Enable Version Control.....
  - 选择Git进行版本控制
- 在项目上右键单击，选择Git--commit directory
  - 将项目提交到本地库
  - 需要输入本地git账号



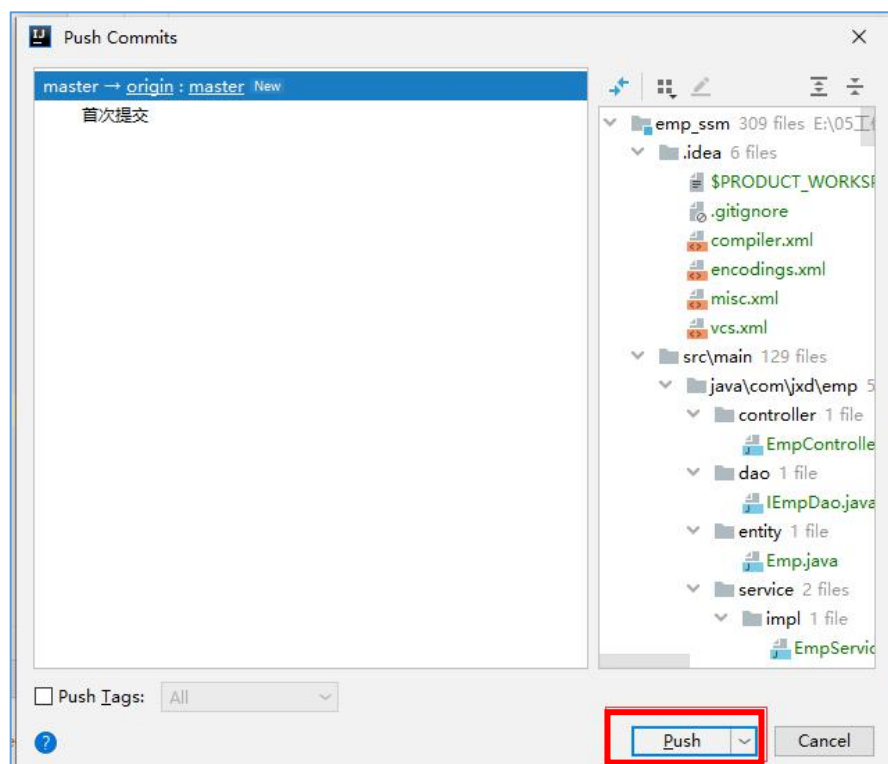
# 将项目push到远程库

- 在项目上右键单击，选择Git--Repository--Push
  - 定义要提交的远程库地址
  - name为远程库的别名



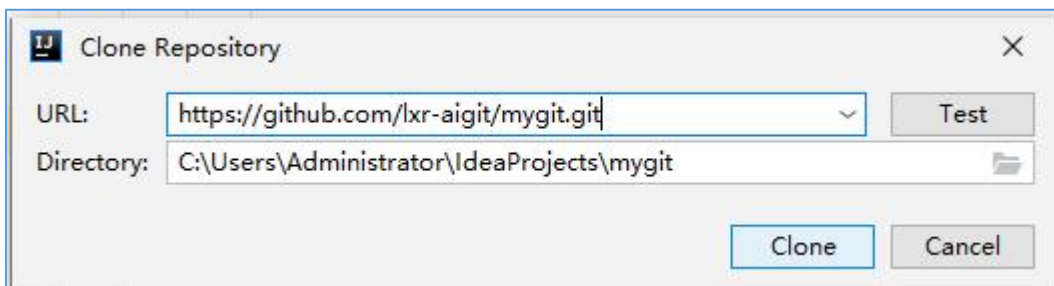
# 将项目push到远程库

- 点击push完成操作
  - 这一步需要输入GitHub账号及密码



# 从远程库拉取代码

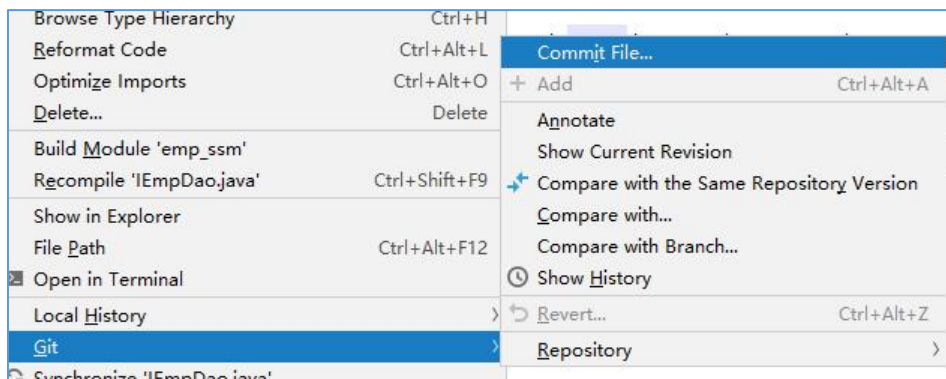
- 成功push之后，其他人可以从远程库clone下来
- 选择File-New-Project from Version Control-Git
  - 输入GitHub远程库地址
  - 设置拉取到的本地仓库地址
  - 点击clone将代码克隆下来
  - 这一步需要输入GitHub账号和密码



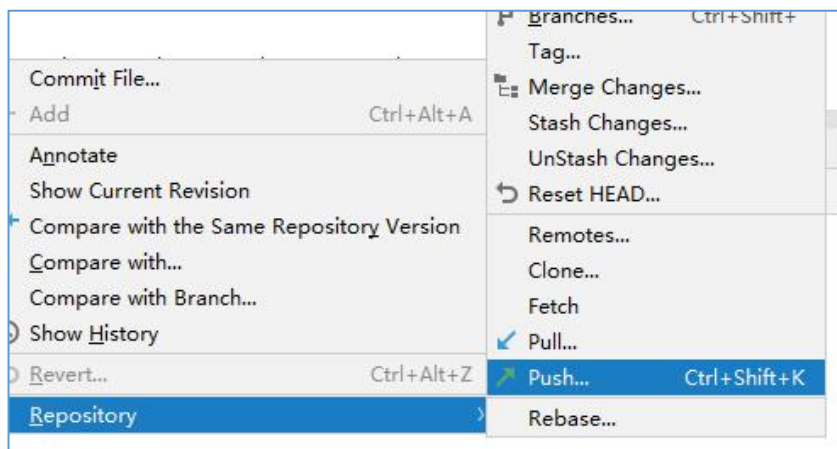


# 修改文件

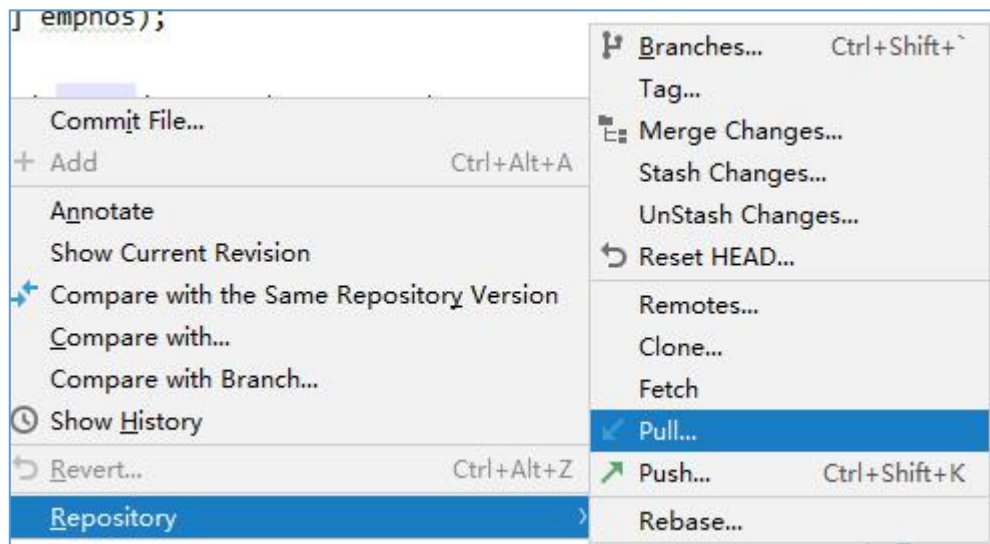
- 当修改文件之后，首先通过Git--commit 操作将其提交到本地库



- 然后在通过Git--Repository--Push将其推送到远程库



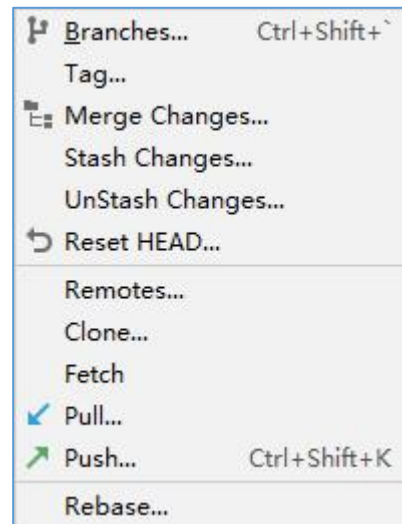
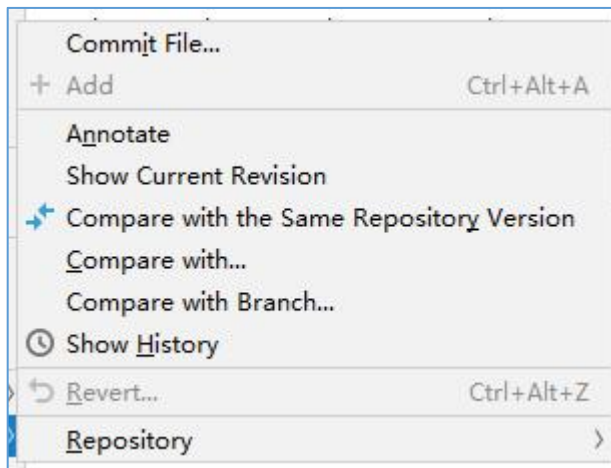
- 通过菜单Git--Repository--Pull可以将远程库的内容更新到本地库



- 其他操作：可以通过点击菜单中其他选项完成相应操作

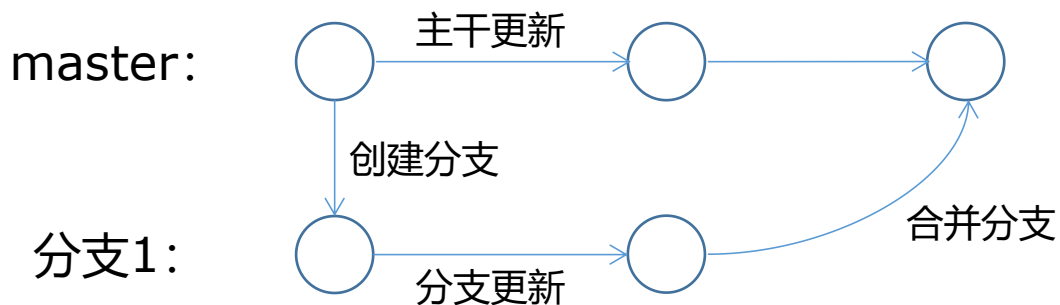
例：Show History 查看历史

Compare with 文件对比



# 分支操作

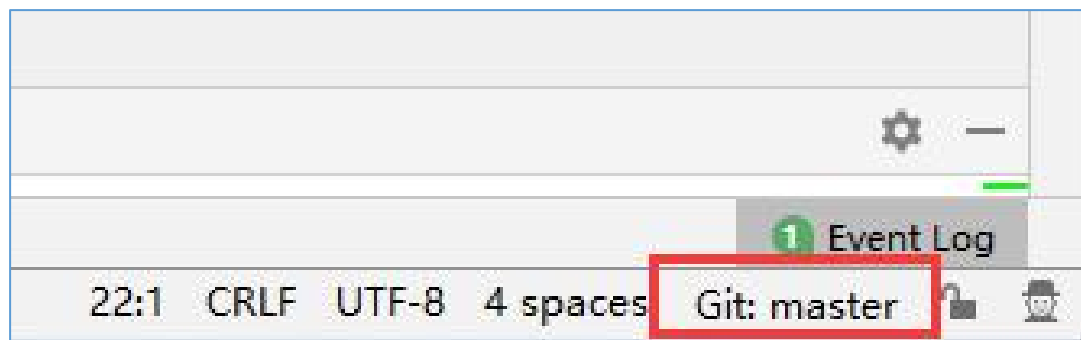
- 什么是分支？
  - 在版本控制的过程中，可以同时推进多个任务，并且多个任务之间互不影响。
  - 如：临时修改bug或者不确定是否加入的新功能，都可以采用分支的方式进行开发。
  - 当分支开发完成确定没有问题之后，可以合并到主分支上。



- 同时并行推进多个功能开发，提高开发效率
- 并行开发过程中，如果一个分支失败，对其他分支没有任何影响
- 失败分支删除后重新开始即可

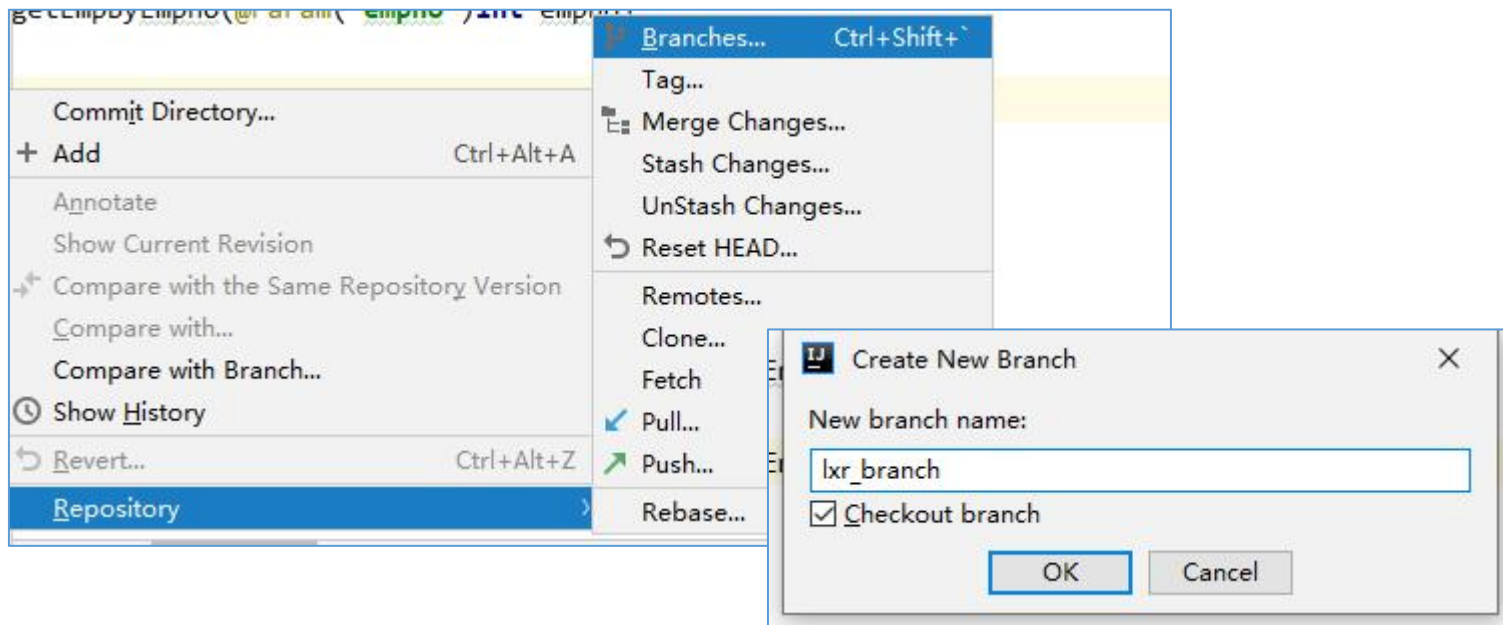
# idea分支应用

- 在idea操作页面的右下角，会显示当前所在分支
- 没有创建其他分支的情况下，都会显示在master主分支



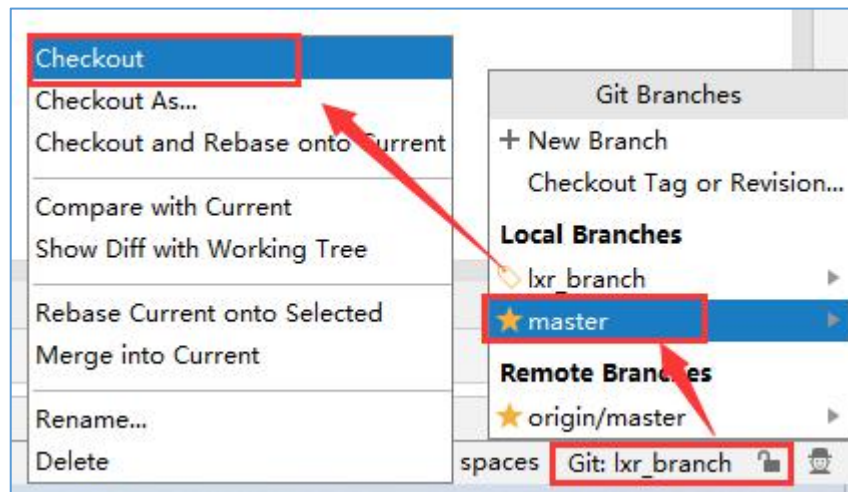
# 创建分支

- 在项目上右键打击，选择Git——Repository——Branches
- 打开新增分支的界面，填写分支名称 lxr\_branch，点击OK即可



# 切换分支

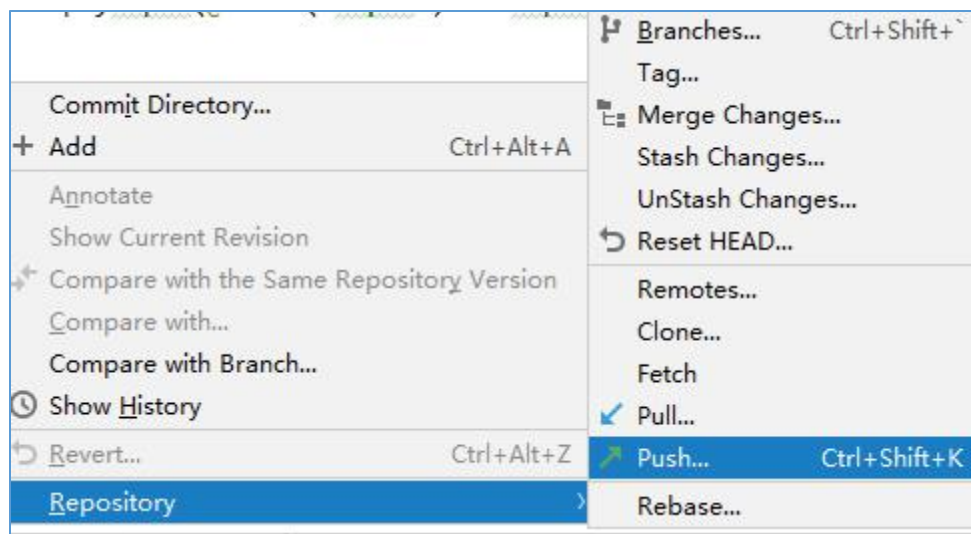
- 创建完成之后，会自动切换到新的分支中
- 新创建的分支内容就是从主分支复制而来的
- 点击当前分支信息可以看到现有的所有分支，通过点击Checkout可以切换到其他分支





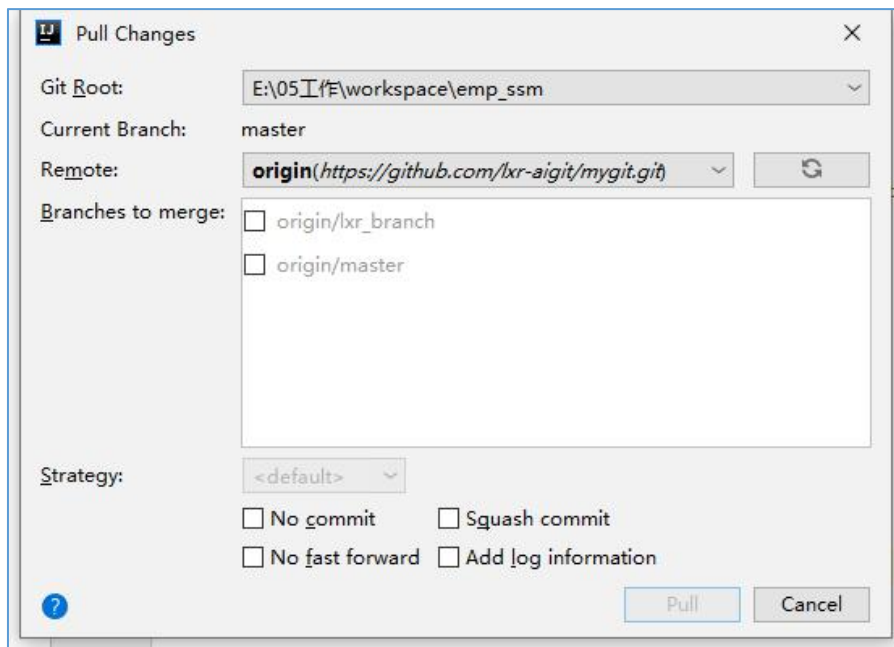
# 将分支内容push到远程库

- 我们在分支上按照功能需求进行独立的开发即可
- push新分支的方式和push主分支的操作完全一样
- 在项目上右键单击选择Git——Repository——push



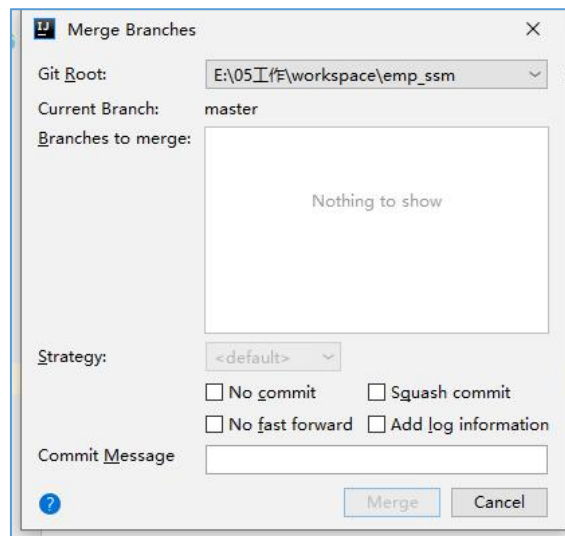
# 从远程库获取分支内容

- 其他人可以从远程库拉取分支内容，右键选择Git——Repository——pull，打开如下界面
- 此时并不是合并操作，所以无需勾选任何分支，直接点击pull即可



# 合并分支到主干

- 当分支功能开发完成之后，就可以将其合并为主分支中
  - 首先切换到主分支master中，然后右键选择菜单Git——Repository——Merge changes
  - 选择要合并的分支，点击Merge即可



# 合并分支时的冲突现象

- 当主分支和其他分支同时修改了某一个文件的同一处内容时，那么合并时就会发生冲突现象，冲突文件中内容会显示一些特殊符号：

```
8 gggggggg
9 <<<<<<< HEAD
10 hhhhhhhh edit by hot_fix
11 =====
12 hhhhhhhh edit by master
13 >>>>>> master
14 iiiiiiii
15 jjjjjjjj
```

当前分支内容

另一分支内容

## 在主分支，通过手动方式修改：

- 1.编辑文件，去掉特殊符号，并将文件修改为正确的内容。
- 2.将文件重新commit，提交到本地库
- 3.再将文件push到远程库

# Git学习资料推荐

---

- B站尚硅谷Git&GitHub培训视频:

<https://www.bilibili.com/video/av24441039?from=search&seid=8031595934221558561>

- 百度网盘资料:

<https://pan.baidu.com/s/1wY4L8HHK38RInfGXtb2GGA>

提取码: ug1v