

Gitee 码云完整使用教程

来源: <https://www.cnblogs.com/jpwz/p/12449391.html>

1. 创建仓库

登录码云 <https://gitee.com/> 创建一个仓库

项目名称

麦兜测试 1. 输入项目名

归属

Re...

路径

https://gitee.com/R... mcdull_test 项目名输入之后自动生成的

项目介绍 非必填

这是一个麦兜用来测试的项目... 2. 项目描述, 随你写不写, 嗯.....还是写点吧

是否开源 3. 选私有共有, 看心情咯

☐ 公开 ☒ 私有

私有项目最多支持 5 人协作 (如拥有多个私有项目, 所有协作人数总计不得超过 5 人)
企业项目, 更适合使用码云企业版, [了解更多 >>](#)

选择语言 4. 选择语言

Objective-C

添加 .gitignore

Xcode 5. 这个好像不选也没关系

☒ 使用Readme文件初始化这个项目
☐ 使用Issue模板文件初始化这个项目 ①
☐ 使用Pull Request模板文件初始化这个项目 ②

6. 随便选个, 一般似乎都选的第一个

导入已有项目

创建 7. 创建

2. 使用 git 在本地初始化

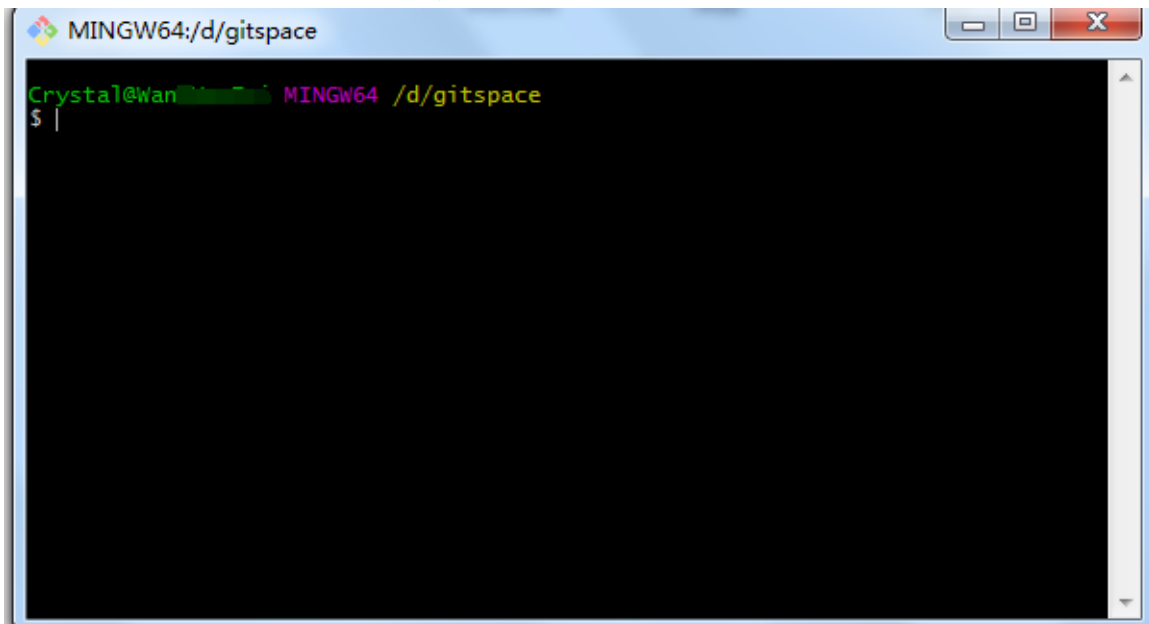
(1) 新建一个目录，存放下载下来的项目，我在 D 盘新建了一个“gitSPACE”文件夹，用来存放下载下来的项目



(2) 进入刚刚新建的文件夹，即进入“gitSPACE”，点击鼠标右键，选择"Git Bash Here", 如下图:



点击“Git Bash Here”之后，可以看到下面界面

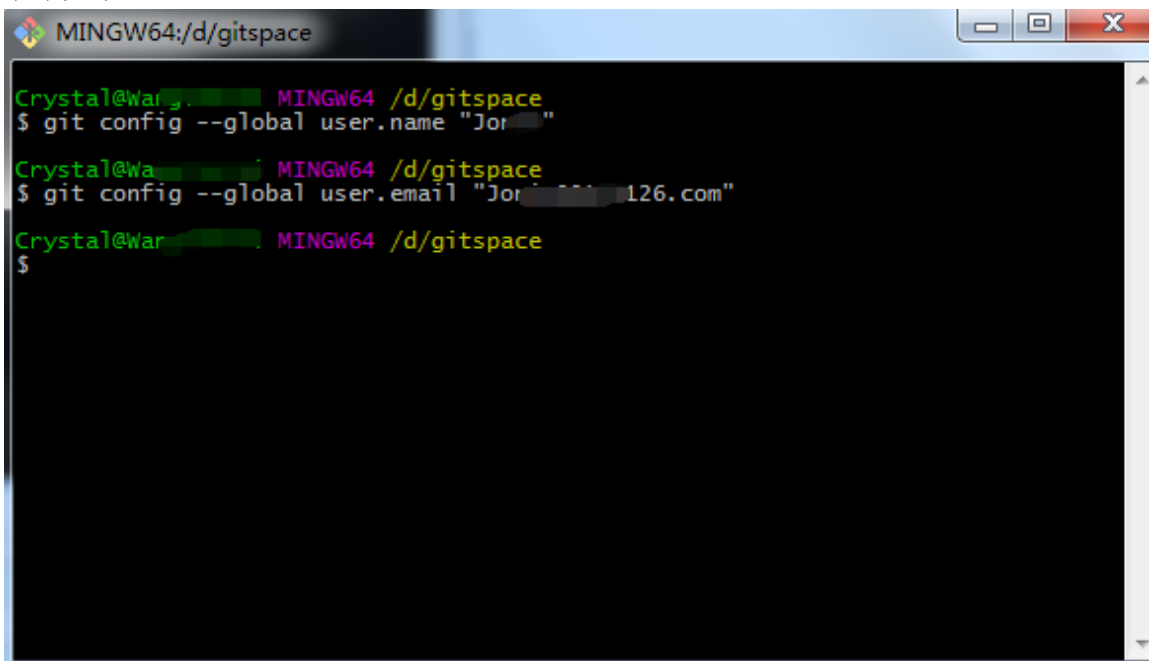


(3) 进行基础配置，作为 git 的基础配置，作用是告诉 git 你是谁，你输入的信息将出现在你创建的提交中，使用下面两条命令：

```
git config --global user.name "你的名字或昵称"
```

```
git config --global user.email "你的邮箱"
```

效果如下：

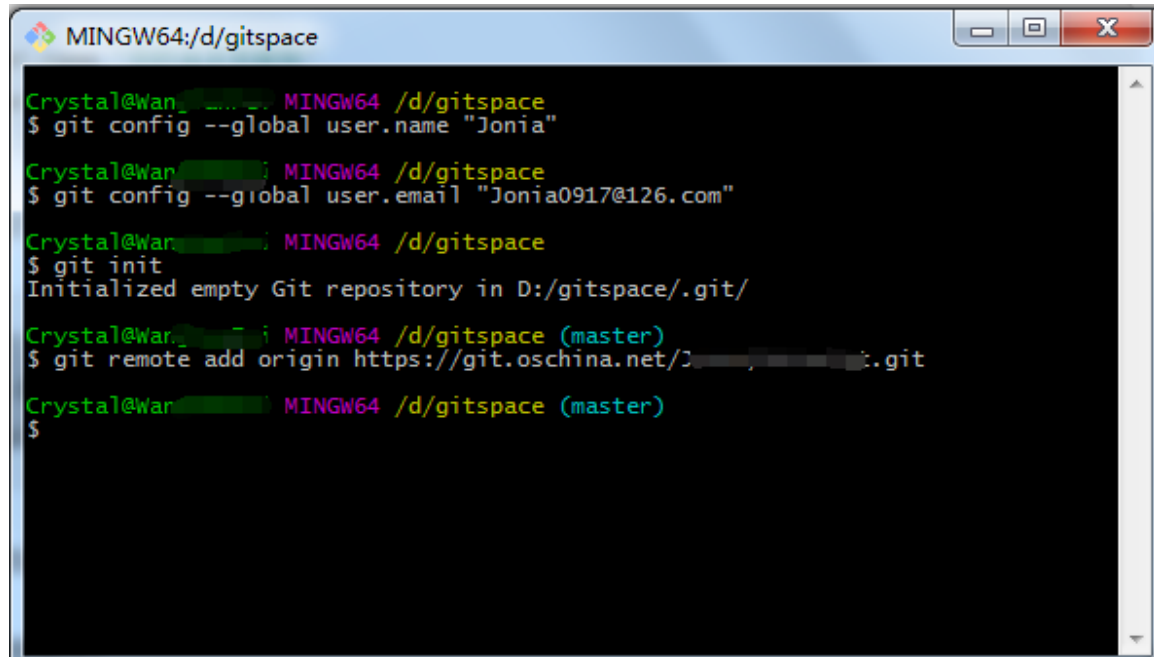


(4) 在 gitSPACE 文件夹中执行下面命令，完成初始化

git init

git remote add origin <你的项目地址> //注:项目地址形式为:https://gitee.com/xxx/xxx.git 或者 git@gitee.com:xxx/xxx.git

效果如下:

A screenshot of a Windows terminal window titled 'MINGW64:/d/gitSPACE'. The window shows a series of git commands being executed. The user sets the global user name to 'Jonia' and the email to 'Jonia0917@126.com'. Then, they initialize a new git repository in the current directory. Finally, they add a remote origin pointing to 'https://git.oschina.net/J...'.

```
Crystal@Wan... MINGW64 /d/gitSPACE
$ git config --global user.name "Jonia"

Crystal@Wan... MINGW64 /d/gitSPACE
$ git config --global user.email "Jonia0917@126.com"

Crystal@Wan... MINGW64 /d/gitSPACE
$ git init
Initialized empty Git repository in D:/gitSPACE/.git/

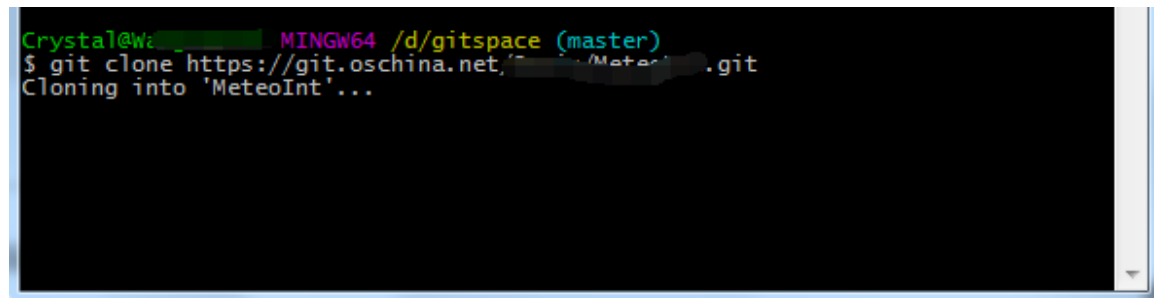
Crystal@Wan... MINGW64 /d/gitSPACE (master)
$ git remote add origin https://git.oschina.net/J... .git

Crystal@Wan... MINGW64 /d/gitSPACE (master)
$
```

(5) 如果你想克隆, 只需要执行命令

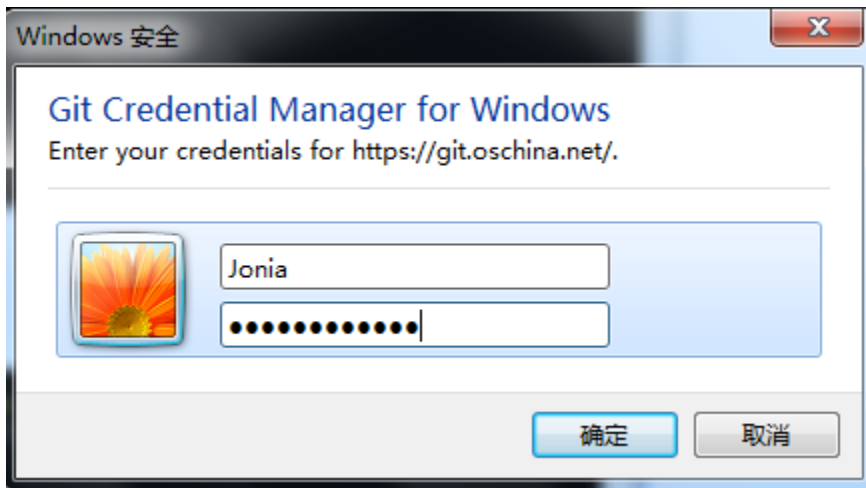
git clone <项目地址>

效果如下:

A screenshot of a Windows terminal window showing the execution of the git clone command. The user clones a repository from 'https://git.oschina.net/J.../MeteoInt.git' into a directory named 'MeteoInt'.

```
Crystal@Wan... MINGW64 /d/gitSPACE (master)
$ git clone https://git.oschina.net/J.../MeteoInt.git
Cloning into 'MeteoInt'...
```

弹出窗口, 输入码云的账户名、密码

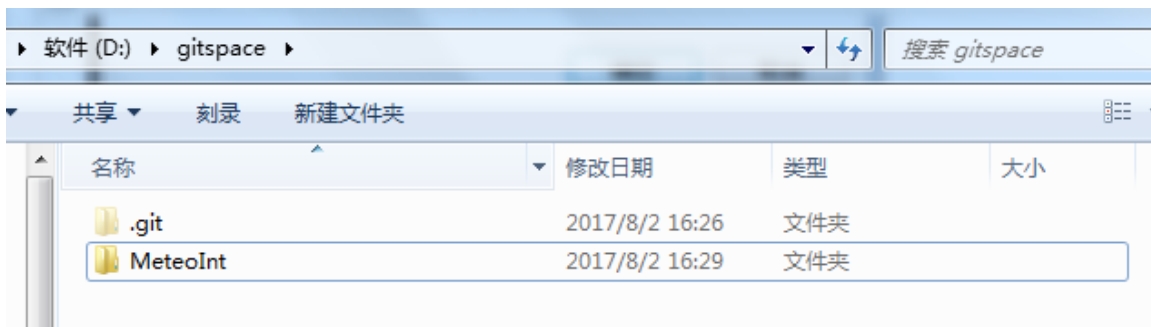


点击“确定”

```
Crystal@wang:~$ i MINGW64 /d/gitSPACE (master)
$ git clone https://git.oschina.net/Int'...
Cloning into 'Int'...
remote: Counting objects: 2943, done.
remote: Compressing objects: 100% (2762/2762), done.
remote: Total 2943 (delta 163), reused 2898 (delta 140)
Receiving objects: 100% (2943/2943), 11.14 MiB | 431.00 KiB/s, done.
Resolving deltas: 100% (163/163), done.
Checking connectivity... done.
Checking out files: 100% (3079/3079), done.

Crystal@wang:~$ i MINGW64 /d/gitSPACE (master)
$
```

再看 gitSPACE 文件夹下，已经下载下来了



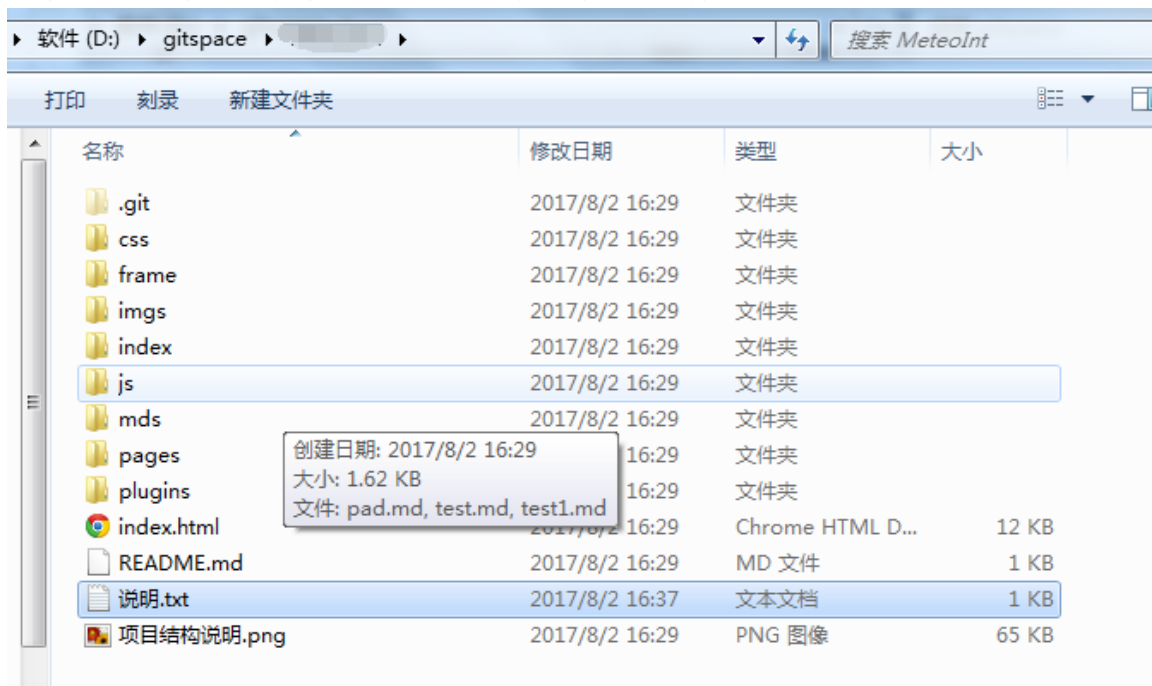
(6) 进入你已经初始化好的或者克隆项目的目录,然后执行:

从服务器下更新项目，因为已经 clone 过，所以不需要再更新

git pull origin master

3.提交项目到码云

在本地项目中做一些修改，比如添加一个"说明.txt"文件



执行下面命令，完成第一次提交

保存到缓存区，或 `git add` 被拖入的项目名

`git add .`

描述这次提交的内容 (推送到本地库中)

`git commit -m "要描述的内容"`

推送到远端仓库码云上，项目大的话，时间会久些

`git push origin master`

注意：如果最后一步报错，可以使用 `git push -f origin master`，来强制覆盖。

`git push origin master //`（正常提交）和

`git push origin master -f //`（强制提交，强制提交可能会把之前的 `commit` 注释信息，不会改变修改的代码，慎用），都是提交到 `master` 分支

```
Crystal@Wan:~$ MINGW64 /d/gitspace (master)
$ git add .

Crystal@Wan:~$ MINGW64 /d/gitspace (master)
$ git commit -m "安装教程测试"
[master (root-commit) af9bf30] 安装教程测试
1 file changed, 1 insertion(+)
create mode 160000 MeteoInt
```

```
Crystal@Wang:~$ git push origin master
Counting objects: 2, done.
Writing objects: 100% (2/2), 211 bytes | 0 bytes/s, done.
Total 2 (delta 0), reused 0 (delta 0)
To https://git.oschina.net/Jonia/MeteoInt.git
+ 9b06ab0...af9bf30 master -> master (forced update)

Crystal@Wang:~$
```

其他常见 git 命令

查看所有分支：git branch -a

切换到某一分支：git checkout 分支名称

合并分支：git merge 原分支 目标分支

4.更新代码到本地

git status（查看本地分支文件信息，确保更新时不产生冲突）

git checkout -- [file name]（若文件有修改，可以还原到最初状态; 若文件需要更新到服务器上，应该先 merge 到服务器，再更新到本地）

git branch（查看当前分支情况）

git checkout [remote branch](#)

git pull

若命令执行成功，则更新代码成功！

可以直接使用：git pull 命令一步更新代码

5.查看和修改用户信息

修改用户名和密码

git config --global user.name "username"

git config --global user.email "email"

查看用户名和密码

git config user.name

git config user.email