

2019

版本控制---github

版本控制 github

王磊

计算机工程学院

CONTENTS



开源社区



申请GitHub账号



登录账号



Git操作

• 开源社区 •

在很长一段时间，项目的源代码是公司的财产，与商业秘密有关，是封闭和不可能公开的。后来开始有人在互联网上分享自己写的代码，互相讨论，逐渐形成了一种特殊的虚拟社区。社区成员大部分都是职业的程序员或编程爱好者，他们根据相应的开源软件许可证协议公布软件源代码，分享源代码，共同自由学习交流。

GitHub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。GitHub于2008年4月10日正式上线。目前注册用户超过350万，拥有超过900万开发者用户。

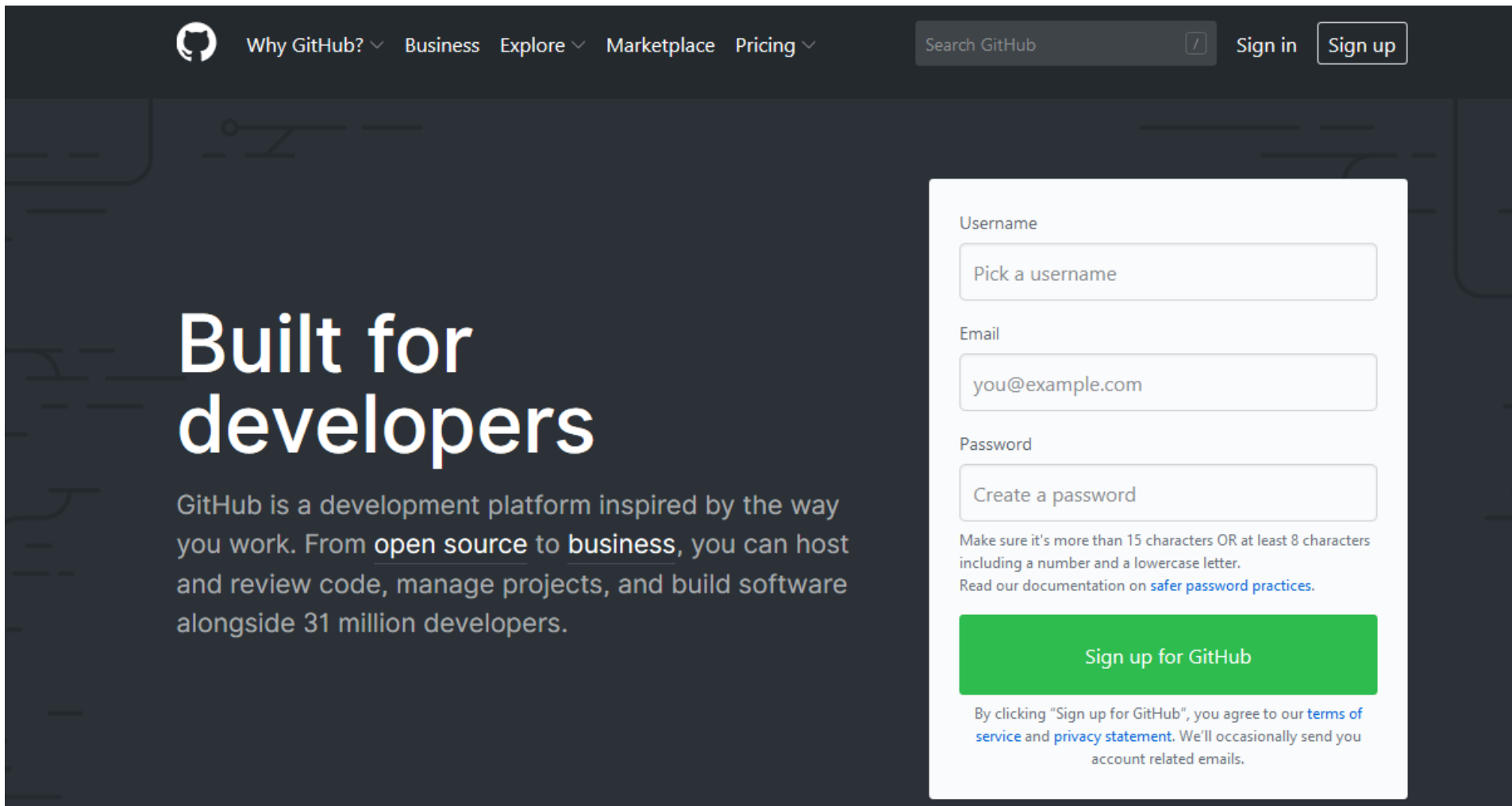
GitHub的出现，极大地推进开源社区的发展。

使用GitHub好处：

- 1、获取最新最热门最实用的开源组件，有助于开发公司项目
- 2、获取最流行的技术相关源代码，有助于参考学习借鉴
- 3、参与感兴趣的开源项目，增强与他人协作开发的能力
- 4、创建属于自己的开源项目，提升编程能力，打造个人名片

一、访问GitHub，申请账号

打开浏览器，在地址栏输入https://github.com



The image shows the GitHub homepage with a dark background. On the left, the text "Built for developers" is prominently displayed in white, followed by a paragraph describing GitHub as a development platform. On the right, a white sign-up form is overlaid. The form contains three input fields: "Username" with the placeholder "Pick a username", "Email" with the placeholder "you@example.com", and "Password" with the placeholder "Create a password". Below the password field, there is a note about password requirements and a link to "safer password practices". A large green button labeled "Sign up for GitHub" is positioned below the form. At the bottom of the form, there is a disclaimer about agreeing to the terms of service and privacy statement.

Why GitHub? ▾ Business Explore ▾ Marketplace Pricing ▾ Search GitHub / Sign in Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 31 million developers.

Username

Email

Password

Make sure it's more than 15 characters OR at least 8 characters including a number and a lowercase letter.
Read our documentation on [safer password practices](#).

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

二、完成申请，验证链接

- 1、显示申请账号的页面，输入你的用户名（Uername）、邮箱地址（Email）和登录密码之后，点击Sign up for GitHub按钮完成申请。
- 2、账号申请成功之后，你的邮箱会收到一封验证邮件，点击验证链接。

Almost done, **@bbcwanglei!** To complete your GitHub sign up, we just need to verify your email address:

bbcwanglei@126.com

Verify email address

Once verified, you can start using all of GitHub's features to explore, build, and share projects.

Button not working? Paste the following link into your browser: https://github.com/users/bbcwanglei/emails/63444901/confirm_verification/c6f8a4ea1134c63cd864b3fff1496d307f56247a

You're receiving this email because you recently created a new GitHub account or added a new email address. If this wasn't you, please ignore this email.


Welcome,
@bbcwanglei!

GitHub lets you version your code, collaborate with others, and share what you've built with the world. We're excited to see what you make.



三、登录账号

输入用户名和密码之后，登录github



Sign in to GitHub

Username or email address

Password [Forgot password?](#)

[Sign in](#)

New to GitHub? [Create an account.](#)

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

[Read the guide](#) [Start a project](#)

[Discover repositories](#)

创建新仓库

已有仓库

Repositories

[New repository](#)

Find a repository...

- [bbcwanglei/mygit](#)
- [bbcwanglei/JavaEE](#)

Browse activity

Discover interesting projects and people to populate your personal news feed.

Your news feed helps you keep up with recent activity on repositories you [watch](#) and people you [follow](#).

[Explore GitHub](#)


• 四、创建新仓库 •

Create a new repository

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 bbcwanglei ▾

Repository name

test ✓

(1)仓库名称

Great repository names are short and memorable. Need inspiration? How about [cautious-barnacle](#).

Description (optional)

test

☒  Public

Anyone can see this repository. You choose who can commit.

(2)仓库必须公开

☐  Private


You choose who can see and commit to this repository.

☒ Initialize this repository with a README

(3)初始化仓库

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾ 

Create repository

(4)开始创建

五、仓库创建成功

仓库创建成功，拥有300M空间。

bbcwanglei / test

Watch

0

Star

0

Fork

0

Code

Issues0

Pull requests0

Projects0

Wiki

Insights

Settings

test

Edit

Manage topics

1 commit

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

bbcwanglei Initial commit

Latest commit 5c4c593 just now

README.md

Initial commit

just now

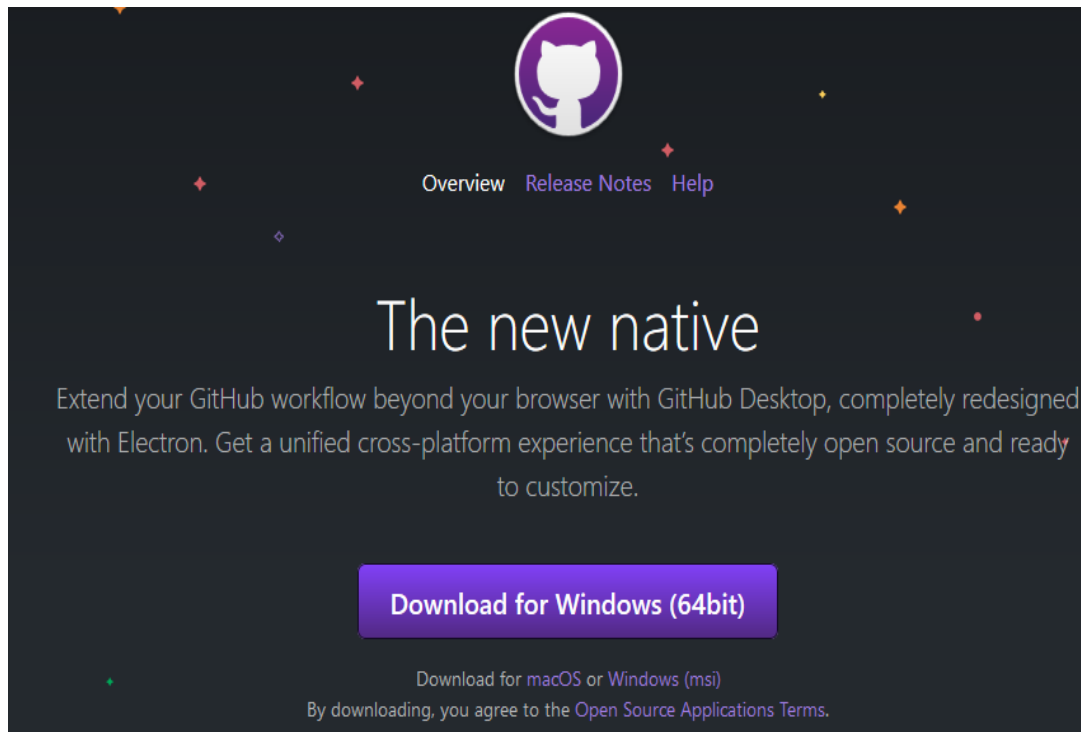
README.md

test

test

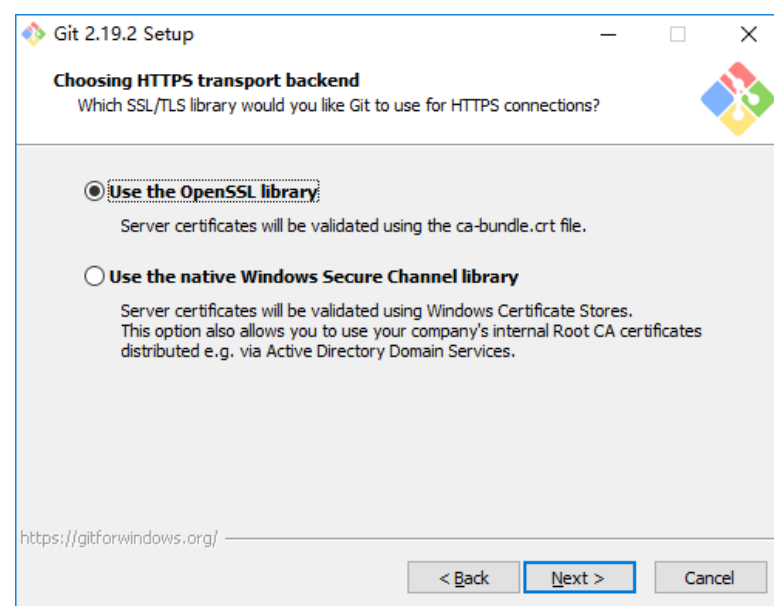
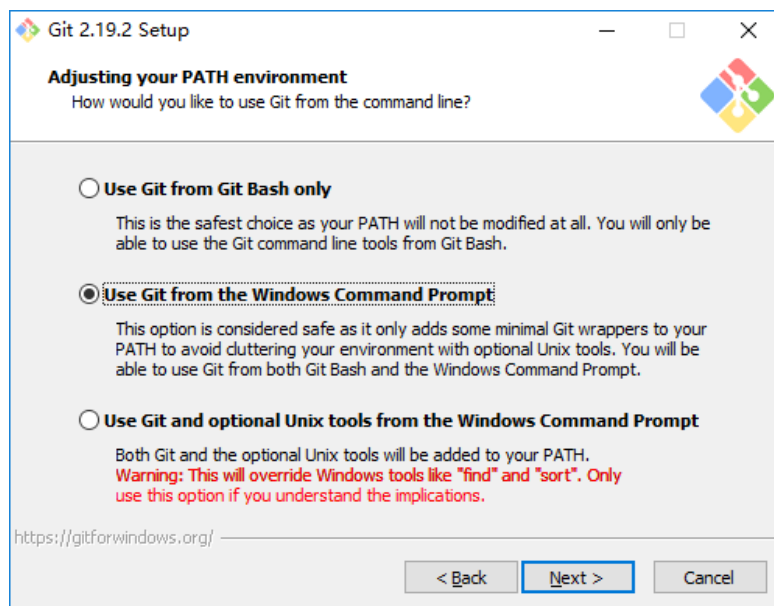
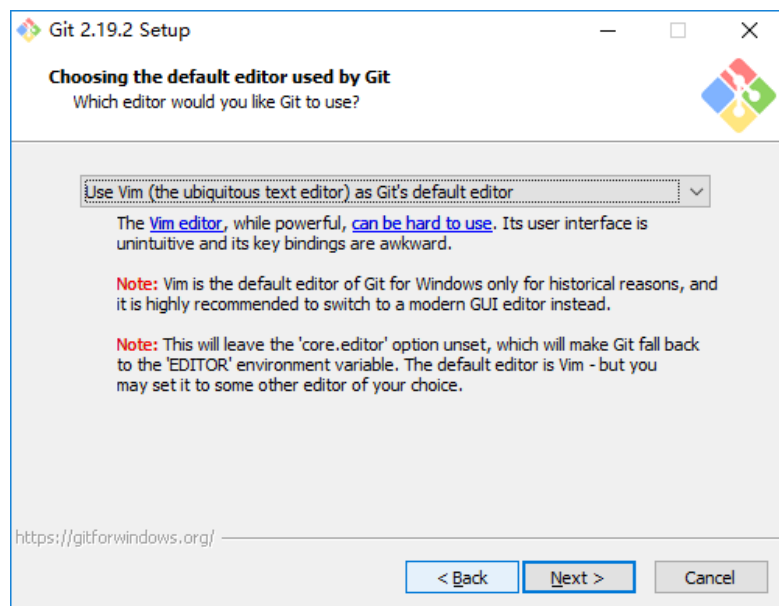
• 六、安装GitBash •

1、下载GitBash并安装:<https://desktop.github.com/>



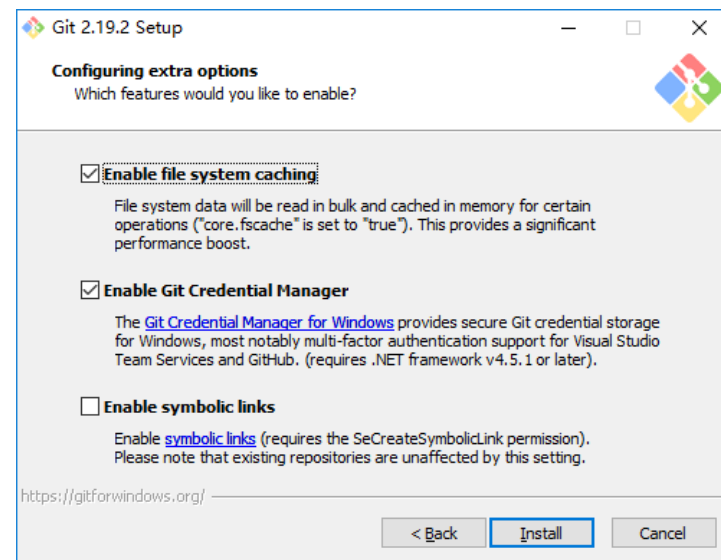
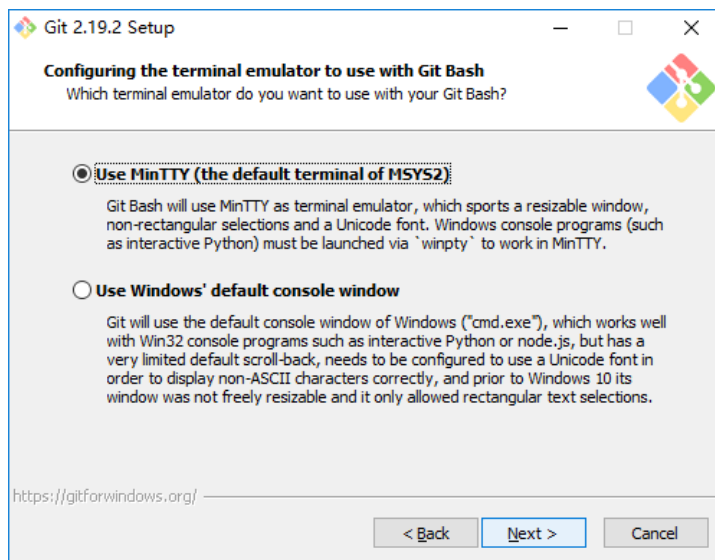
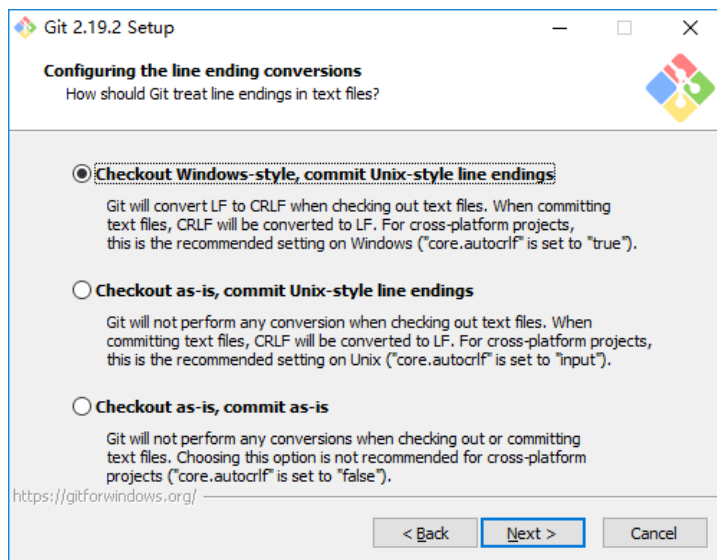
六、安装GitBash

1、下载GitBash并安装:<https://desktop.github.com/>



• 六、安装GitBash •

1、下载GitBash并安装:<https://desktop.github.com/>



• 七、生成SSH密钥 •

1、生成SSH密钥

打开GitBash，使用命令生成ssh key

`ssh-keygen -t rsa -C "your_email@example.com"`

将密钥拷贝到GitHub，**公钥上传，私钥自己保留。**

```
MINGW64:/c/Users/Administrator
Administrator@MINGW64 ~
$ ssh-keygen.exe -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Administrator/.ssh/id_rsa):
Created directory '/c/Users/Administrator/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Administrator/.ssh/id_rsa.
Your public key has been saved in /c/Users/Administrator/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Q6t4QInoaNQavumJlAacM1EQHXNozk9ctFEzakrAfx8 Administrator@
The key's randomart image is:
+---[RSA 2048]-----+
| o*=o..o.+          |
| o.*+. .+. o        |
| ooB.=...+          |
| *=o+. +oo E        |
| o=+ +.. S .        |
|.+. +. o            |
| oo . o              |
| o.. .               |
| ...                 |
+---[SHA256]-----+
Administrator@MINGW64 ~
$
```

```
ws10 (C:) > 用户 > Administrator > .ssh

名称
^

id_rsa
id_rsa.pub

id_rsa.pub - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBA
+/ZgjZXHv8fJIEM6zfdSybx7F8/eg5QH5eE1E
+quHCMXtrQvMxma9FhwOd87/p9AzwEDrPU5NC
+auql6TonU/tla6o37yw2Eut/a/Lij5XvcXDv
5t25DD6LfiI1Wh7YHD/edYFbDmh/ Administ
```

八、clone代码到本地

2、将~/.ssh/id_rsa.pub密钥添加到GitHub的密钥管理

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Sessions

Blocked users

Repositories

Organizations


Saved replies

Applications

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



SSH

debian_virtual_win7_sshkey_jdxwl@126.com

1c:20:97:1a:91:ea:e6:00:0d:99:c7:94:65:18:84:54

Added on 26 Nov 2018

Never used — Read/write

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

GPG keys

New GPG key

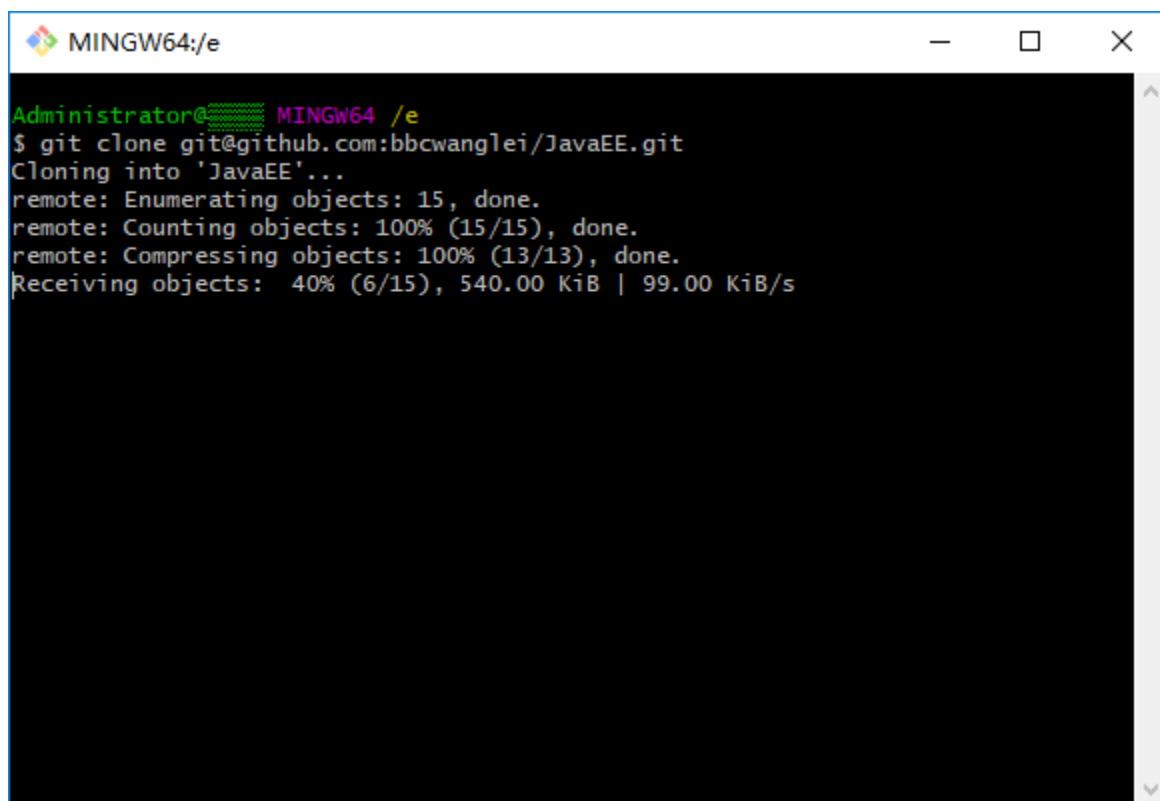
There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).






九、开始clone

1、使用cd命令进入到相应的目录，然后使用如下命令进行clone:

```
$ git clone git@github.com:bbcwanglei/JavaEE.git
```



```
Administrator@MINGW64 /e
$ git clone git@github.com:bbcwanglei/JavaEE.git
Cloning into 'JavaEE'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (13/13), done.
Receiving objects: 40% (6/15), 540.00 KiB | 99.00 KiB/s
```

-  1JavaEE培训@环境搭建、Servlet、连接数据库.pdf
-  2struts2培训@part1@搭建Struts2开发环境.pdf
-  3struts2培训@part2@拦截器、通配符、OGNL.pdf
-  README.md
-  xkuser.rar

• 十、push代码到GitHub •

- 1、对源代码做出增加、删除或者修改
- 2、也可以新增一些其他文件
- 3、使用以下命令进行提交：

3.1 首先注册身份：

```
git config --global user.email "jdxwl@126.com"
```

```
git config --global user.name "wl"
```

3.2 新增变化到当前git工作区

```
git add .
```

3.3 把最新版本内容放至暂存区

```
git commit -m "xxx"
```

3.4 向远程GitHub提交

```
git push
```

• 十、push代码到GitHub •

4、观察执行结果

```
Administrator@MINGW64 /e/JavaEE (master)
$ git commit -m "add 2 file by wl"
[master e7d0cd2] add 2 file by wl
4 files changed, 138 insertions(+)
create mode 100644 "linux\345\221\275\344\273\244.txt"
create mode 100644 "web \345\211\215\347\253\257 \345\255\246\344\271\240 \350\267\257\347\272\277.mm"
delete mode 100644 xkuser.rar
create mode 100644 "\345\255\246\344\271\240bootstrap.png"

Administrator@MINGW64 /e/JavaEE (master)
$ |
```

```
Administrator@MINGW64 /e/JavaEE (master)
$ git push
Warning: Permanently added the RSA host key for IP address '13.250.177.223' to the list of known hosts.
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 27.13 KiB | 4.52 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:bbcwanglei/JavaEE.git
   ad2873a..e7d0cd2  master -> master

Administrator@MINGW64 /e/JavaEE (master)
$
```


十一、观察结果

5、GitHub远程仓库已经发生变化



bbcwanglei add 2 file by wl

1JavaEE培训@环境搭建、Servlet、连接数据库.pdf

2struts2培训@part1@搭建Struts2开发环境.pdf

3struts2培训@part2@拦截器、通配符、OGNL.pdf

README.md

linux命令.txt

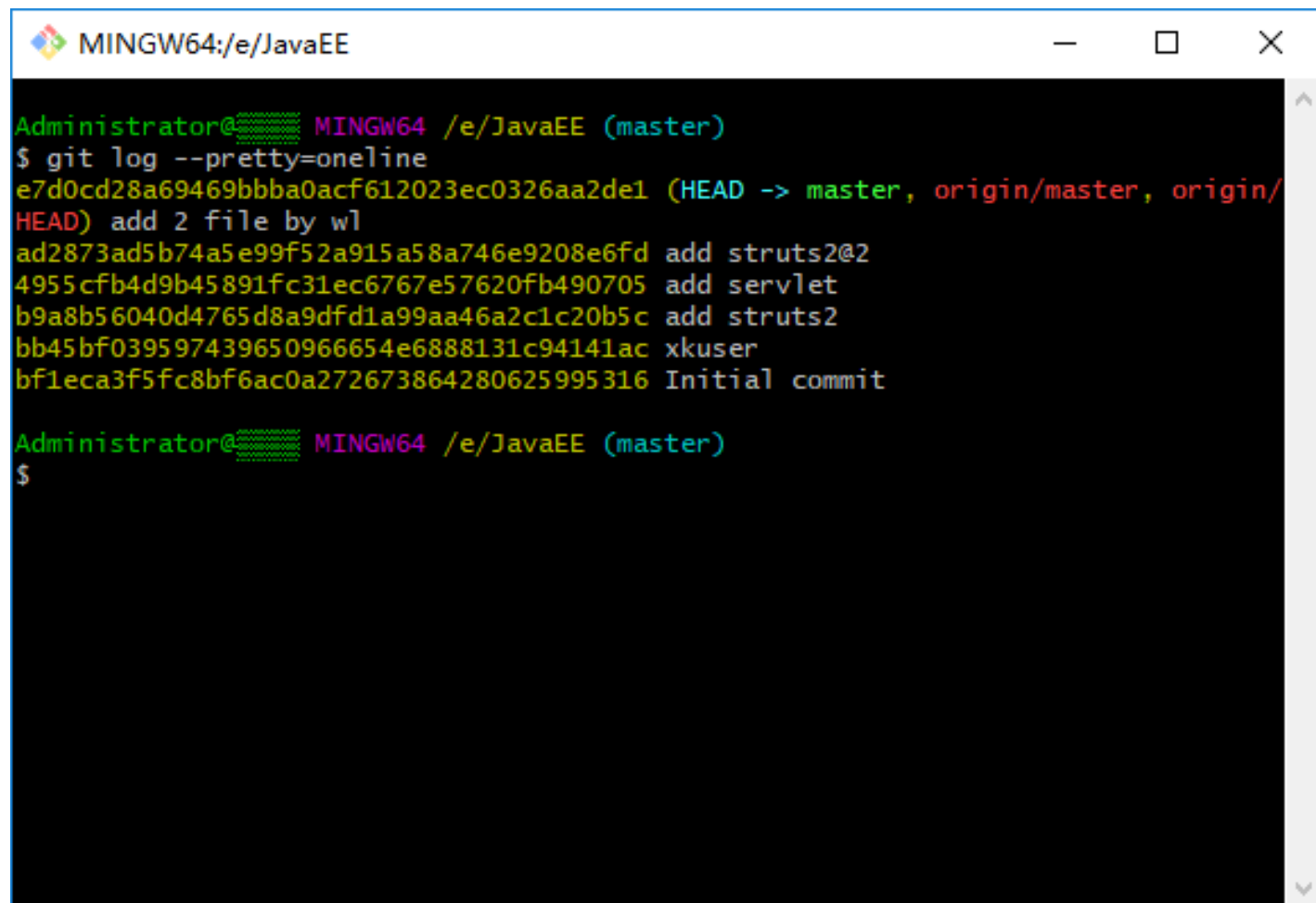
web 前端 学习 路线.mm

学习bootstrap.png

十二、查询历史

查询历史不同版本ID号

```
git log --pretty=oneline
```



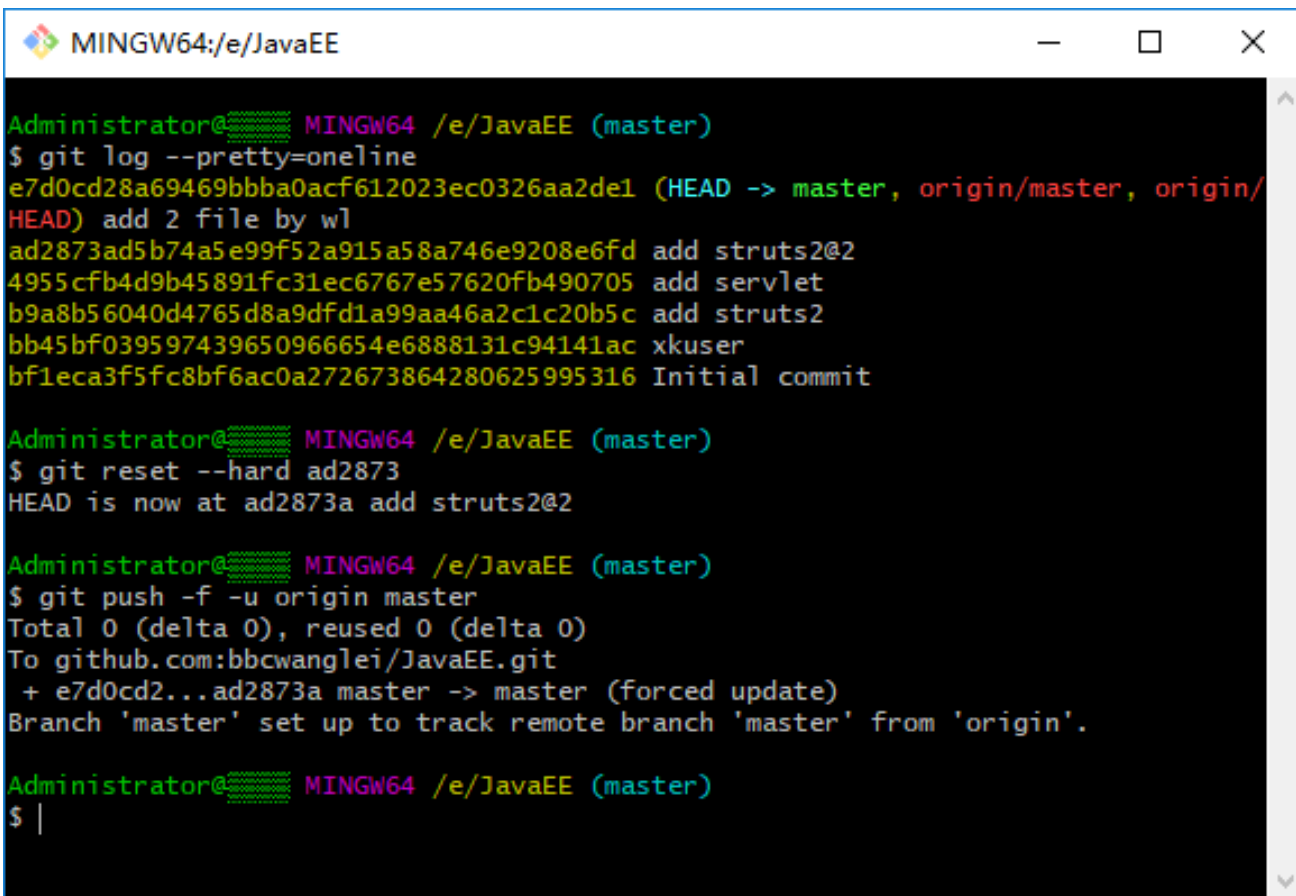
```
Administrator@MINGW64:/e/JavaEE (master)
$ git log --pretty=oneline
e7d0cd28a69469bbba0acf612023ec0326aa2de1 (HEAD -> master, origin/master, origin/HEAD) add 2 file by w1
ad2873ad5b74a5e99f52a915a58a746e9208e6fd add struts2@2
4955cfb4d9b45891fc31ec6767e57620fb490705 add servlet
b9a8b56040d4765d8a9dfd1a99aa46a2c1c20b5c add struts2
bb45bf039597439650966654e6888131c94141ac xkuser
bf1eca3f5fc8bf6ac0a272673864280625995316 Initial commit

Administrator@MINGW64:/e/JavaEE (master)
$
```

• 十三、恢复到历史版本 •

恢复到历史版本

- (1) `git reset --hard 版本ID` --移动版本指针
- (2) `git push -f -u origin master` --强制上传到远程，并覆盖



```
Administrator@MINGW64: /e/JavaEE (master)
$ git log --pretty=oneline
e7d0cd28a69469bbb0acf612023ec0326aa2de1 (HEAD -> master, origin/master, origin/HEAD) add 2 file by wl
ad2873ad5b74a5e99f52a915a58a746e9208e6fd add struts2@2
4955cfb4d9b45891fc31ec6767e57620fb490705 add servlet
b9a8b56040d4765d8a9dfd1a99aa46a2c1c20b5c add struts2
bb45bf039597439650966654e6888131c94141ac xkuser
bf1eca3f5fc8bf6ac0a272673864280625995316 Initial commit

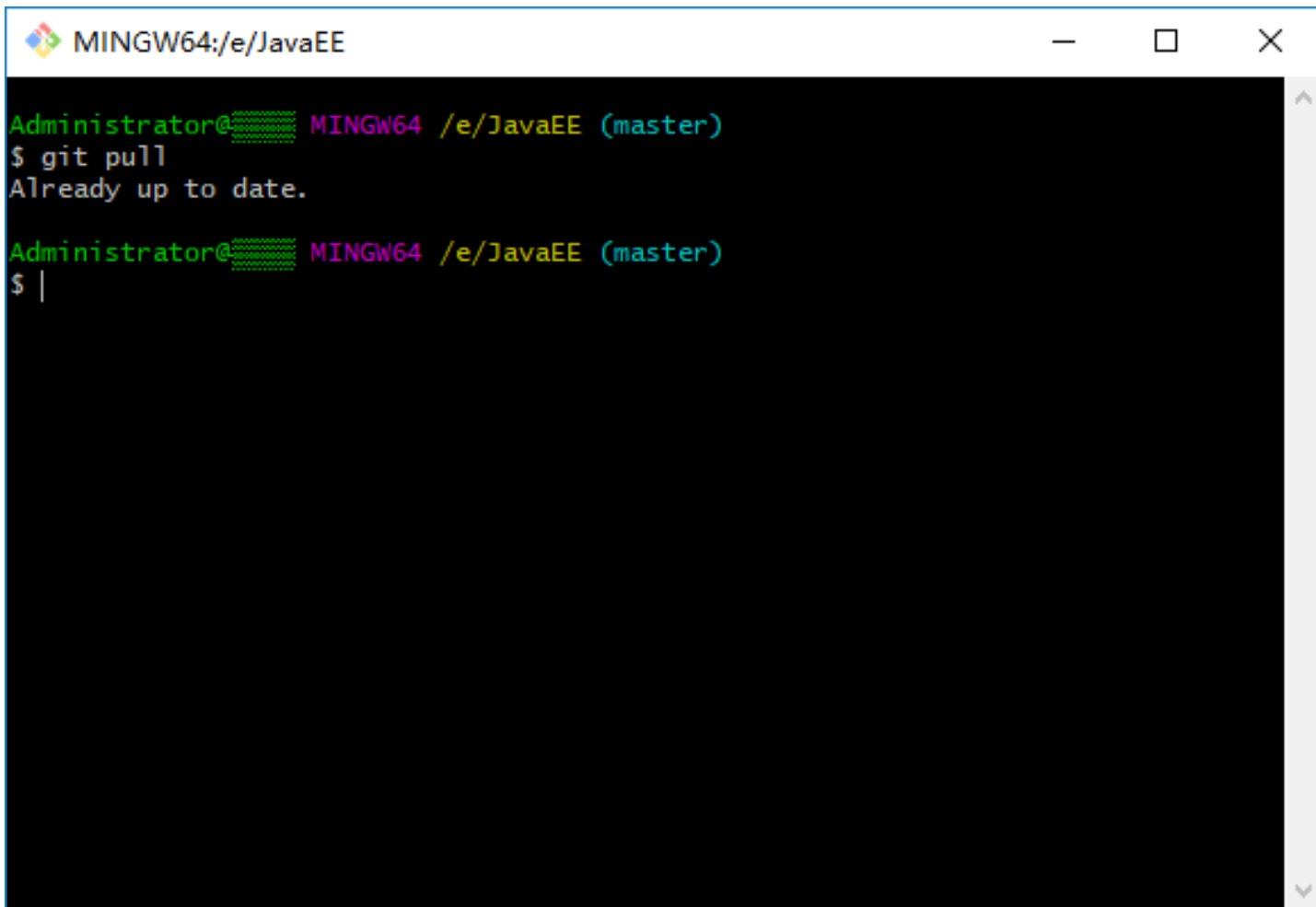
Administrator@MINGW64: /e/JavaEE (master)
$ git reset --hard ad2873
HEAD is now at ad2873a add struts2@2

Administrator@MINGW64: /e/JavaEE (master)
$ git push -f -u origin master
Total 0 (delta 0), reused 0 (delta 0)
To github.com:bbcwanglei/JavaEE.git
+ e7d0cd2...ad2873a master -> master (forced update)
Branch 'master' set up to track remote branch 'master' from 'origin'.

Administrator@MINGW64: /e/JavaEE (master)
$ |
```

十四、重新pull最新版本

本地仓库重新pull最新版本
git pull

A screenshot of a Windows terminal window titled 'MINGW64:/e/JavaEE'. The window shows a command prompt session where the user runs 'git pull'. The output indicates that the local repository is already up to date. The prompt then returns to the user, who has entered a vertical bar character '|'.

```
MINGW64:/e/JavaEE
Administrator@MINGW64 /e/JavaEE (master)
$ git pull
Already up to date.
Administrator@MINGW64 /e/JavaEE (master)
$ |
```

• 十五、只删除远程仓库，不删除本地仓库 •

```
$ git pull origin master // 将项目从远程库拉下来
```

```
$ git rm -r --cached 1.txt // 删除指定文件，多个文件之间空格隔开
```

```
$ git commit -m '删除1.txt'
```

```
$ git push -u origin master
```

• 十六、同时删除Github本地库和远程仓库的文件 •

进入本地文件夹中相应的文件夹，删除需要删除的文件/文件夹，执行以下命令：

```
$ git add *
```

```
$ git commit -m 'delete file'
```

```
$ git push origin master
```