

一、 Windows 需安装 GitBash(下载后一路下一步即可安装成功)

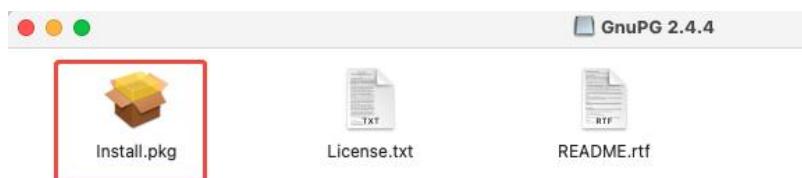
<http://192.144.220.147:9000/bin/Git-2.42.0.2-64-bit.exe>

二、 Mac 需要安装 gunpg

1、 gunpg 下载

<http://192.144.220.147:9000/bin/GnuPG-2.4.4.dmg>

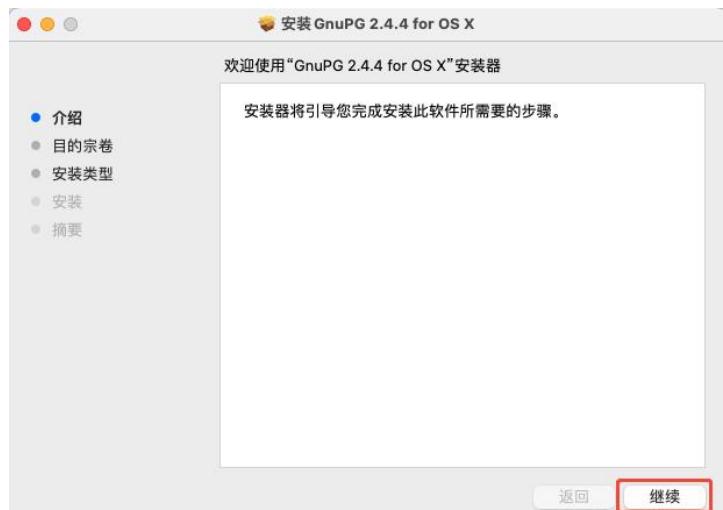
2、 打开如下图



3、 如遇阻止，请按下图操作



4、 接下来一路下一步



三、生成密钥

```
gpg --quick-generate-key 改为你的邮箱@duole.com rsa2048 sign,encrypt,auth 0y
```

```
$ gpg --quick-generate-key yangtianfang@duole.com rsa2048 sign,encrypt,auth 0y
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key 0600813F118698C4 marked as ultimately trusted
gpg: directory '/c/Users/younger/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/c/Users/younger/.gnupg/openpgp-revocs.d/B526587B7DA616034617
64030600813F118698C4.rev'
public and secret key created and signed.

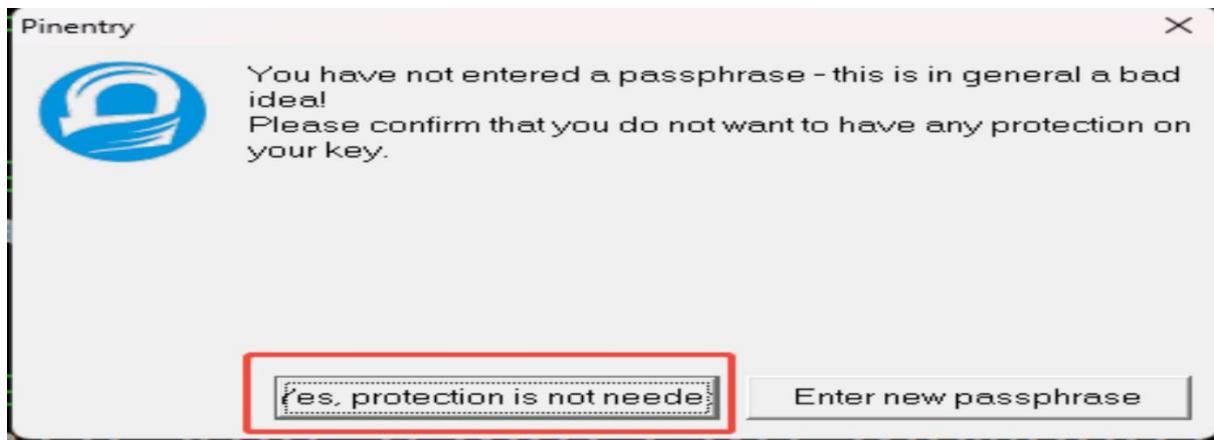
pub    rsa2048 2024-12-28 [SCEA]
      B526587B7DA61603461764030600813F118698C4
uid            yangtianfang@duole.com
```

tips:密码这里建议为空

```
younger@LAPTOP-SI4CG:~$ gpg -k
Pinentry
Please enter the passphrase to
protect your new key

younger@LAPTOP-SI4CG:~$ gpg --quick-genera... e.com 2048 0
gpg: Key generation

younger@LAPTOP-SI4CG:~$ gpg --quick-generate-key yangtianfang@duole.com rsa2048 sign,encrypt,auth 0y
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
```



四、查看密钥

```
gpg --list-keys
```

```
$ gpg --list-keys  
/c/Users/younger/.gnupg/pubring.kbx  
-----  
pub rsa2048 2024-12-28 [SCEA]  
      B526587B7DA61603461764030600813F118698C4  
uid          [ultimate] yangtianfang@duole.com
```

五、导出公钥

```
gpg --export --armor 改为你的邮箱@duole.com
```

```
$ gpg --export --armor yangtianfang@duole.com  
-----BEGIN PGP PUBLIC KEY BLOCK-----  
  
mQENBGdvjBQBCADhDLFshsASCUT+CXTHcNRlMFfIwfC8GP8AEchrUjWOIX6G7pGr  
5SecAlSz2ZEr7n4RCxHP3glhcksuHK9gTZ090rHgDXum5EITt76eCmJAnQyAZY8I  
rF4jQou... 4rUE0adT1yn4Wibty10ha4yLRfojCIU2Ed  
As9p7zMnrojN606x7w0... 5ihkfrs01iqwRI8+HUK0uwqK  
URIMmCbpH2... vOZ7Om0MLp/  
G1cALyUCq... 1hbm... z0bk...  
dw9sZS5jb... ,aYXA... Z?... =AIb  
LwULCQgHAgY... YxKTbB... 3813K  
FgDi033rTUVA... PDE7rb5  
ahA34t+cwe... 7tkcTS... fqiyaajjPV2  
6G1oGoweAWC0uPr69WWA+c... IQmi/sxId7dzvb1RhTG0tjy7sjqvM7w2k13WAHD...  
E6UopMPnIKdc2... i9nh9I+0gjtV6Umw8d9XCHBpMr48gFwhn  
zotC99CgR06y6D1A2ZJg1... 1AxbvXJaKMZ5kLAkenJX7ImIZTv7tvZ  
sqfGkUqHupXW  
=OXov  
-----END PGP PUBLIC KEY BLOCK-----
```

六、打开 gitlab 网站，添加密钥

The screenshot shows the GitLab user settings interface. On the left sidebar, under 'GPG密钥' (GPG Key), there is a red arrow pointing to the '偏好设置' (Preferences) link. The main content area is titled 'GPG密钥' (GPG Key) and contains the message 'GPG密钥允许您对签名的提交进行验证。' (GPG keys allow you to verify signed commits). Below this, it says '您的 GPG 密钥 0' (Your GPG key 0) and '没有可访问您的账户的 GPG 密钥。' (No GPG keys are accessible from your account). A red arrow points to the '添加新密钥' (Add new key) button.

The screenshot shows the '添加 GPG 密钥' (Add GPG Key) form. It includes instructions: '添加 GPG 密钥以安全访问 GitLab. 了解更多.' (Add GPG keys for secure access to GitLab. Learn more.) and '密钥' (Key). The key content is shown in a code block:
-----BEGIN PGP PUBLIC KEY BLOCK-----
mQENBGdyjBQBCADhDLfShsAS...
-----END PGP PUBLIC KEY BLOCK-----
A red box highlights the entire key content, and a red arrow points to the '添加密钥' (Add key) button. To the right, the text '如图所示，开始至结尾全部黏贴' (As shown in the figure, paste from start to end) is displayed.

The screenshot shows the user settings page again. The 'GPG密钥' (GPG Key) section now lists a key: '8526587870d61683461764038608813F118698C4' with status '已激活' (Active). A red box highlights the '已激活' status, and a red arrow points to the '操作' (Operations) column where a trash can icon is located. A red message at the bottom says '绿了代表已经成功啦，你好棒呀，下一步等你哟' (Green means it's successful, you're great, next step awaits you).

七、接下来我们在返回客户端进行调配哈

tips:以下修改全局生效（如想单个仓库生效请在项目目录中执行把--global 改为--local）

```
git config --global user.email 改为你的邮箱@duole.com
```

```
git config --global user.signingkey 改为你的邮箱@duole.com
```

```
git config --global commit.gpgsign true
```

```

younger@LAPTOP-SI4CGGRB MINGW64 ~/Desktop/yunwei (master)
$ git config --global user.email yangtianfang@duole.com

younger@LAPTOP-SI4CGGRB MINGW64 ~/Desktop/yunwei (master)
$ git config --global user.signingkey yangtianfang@duole.com

younger@LAPTOP-SI4CGGRB MINGW64 ~/Desktop/yunwei (master)
$ git config --global commit.gpgsign true

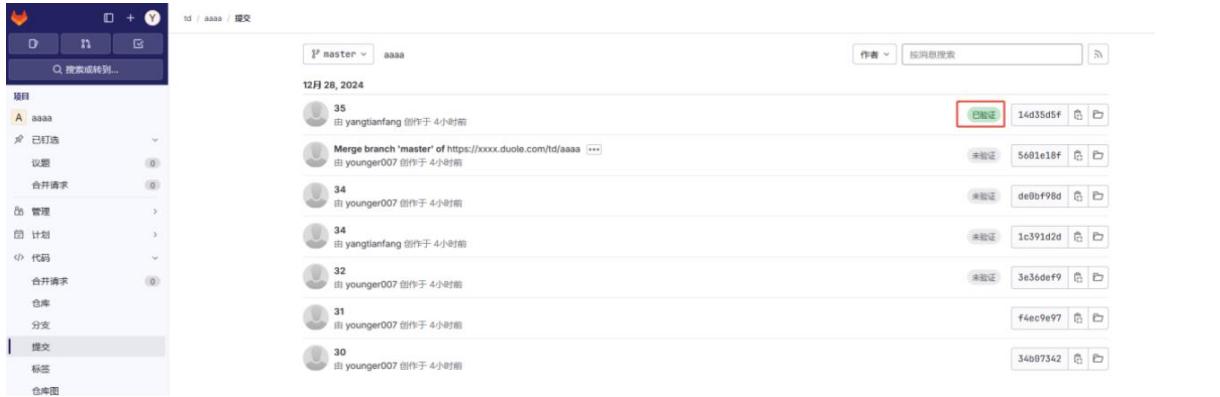
```

八、验证结果

验证配置是否成功，可以检查以下位置的状态



The screenshot shows a GitHub repository named 'aaaa'. A commit by 'yangtianfang' is highlighted with a red box around the '已验证' (Verified) status badge. The commit message is 'Merge branch 'master' of https://xxxx.duole.com/td/aaaa'. The commit hash is '14d35d5f'.



The screenshot shows the commit history for the 'aaaa' repository. It displays several commits, all of which have been signed, as indicated by the '已验证' (Verified) status badges next to their commit messages. The commits are listed with their authors, dates, and hashes.