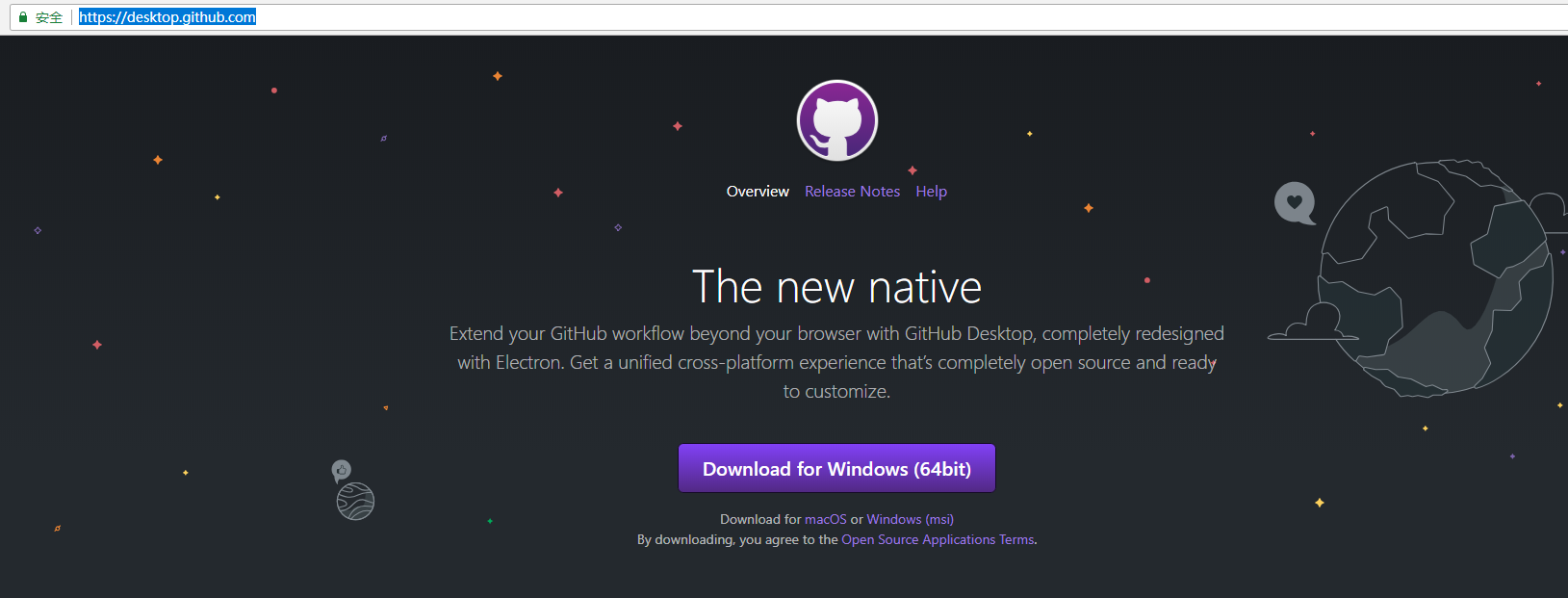
# 注册github账号和安装github客户端

1. <https://github.com/>注册一个github账号
2. Windows安装github客户端

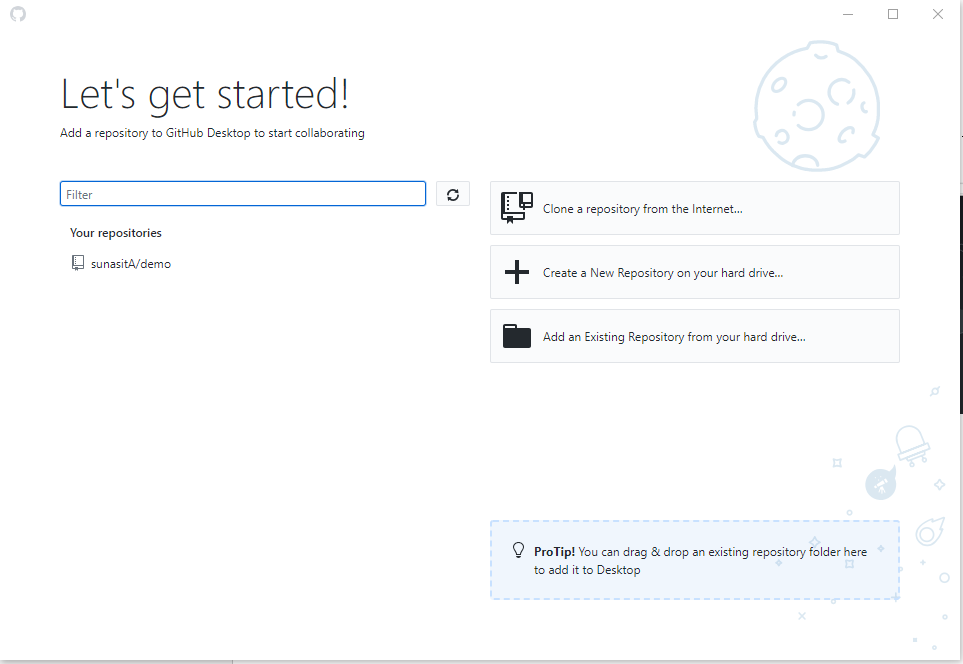
到<https://desktop.github.com/> 点击下图的Download for Windows。GitHubDesktopSetup.exe也上传到了本仓库。



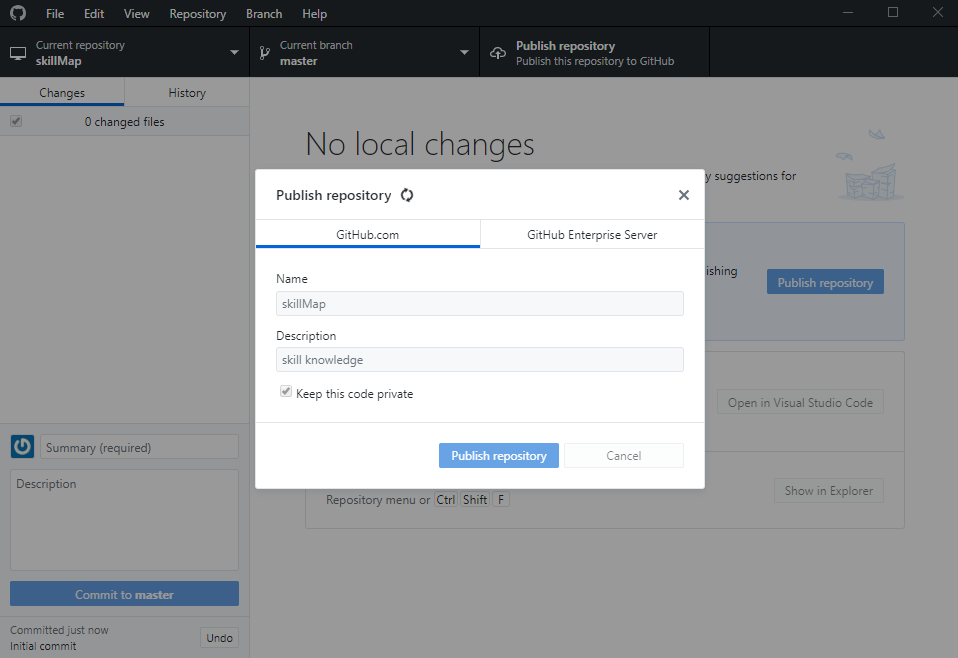
1. 点击exe文件安装，填入1中注册的github账号。

# 创建一个仓库

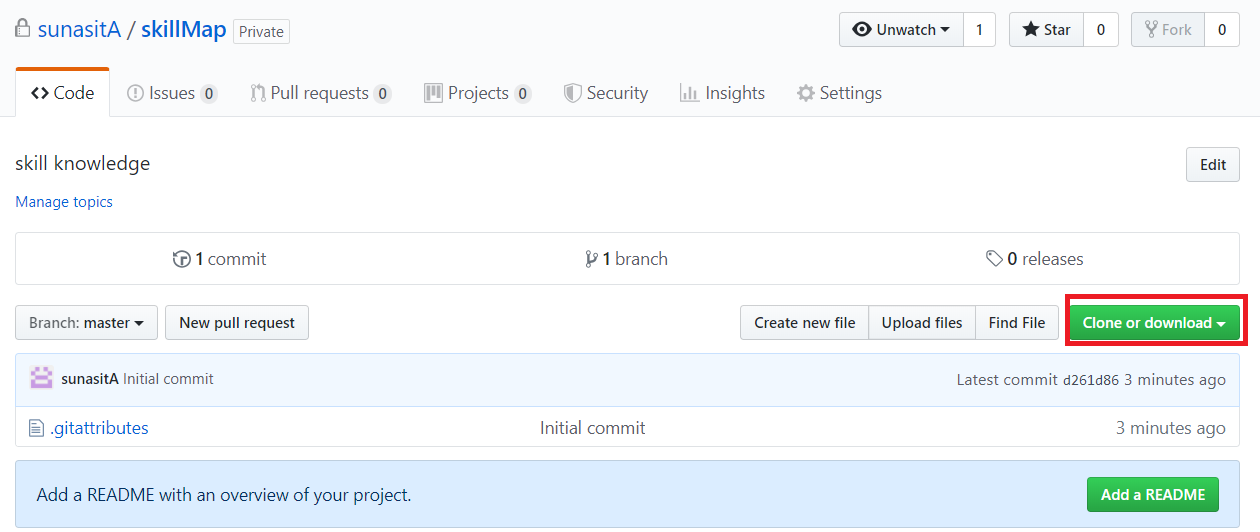
1. 安装完成可见下画面（其中有个项目是我在github web页面创建的，请忽略）



1. 点击4图的 create a new repository on your hard drive 创建一个新仓库skillMap，然后publish



1. 登陆github web界面可以看到新建的仓库，点击红框按钮拿到仓库地址



# Git安装和项目clone

1. Windows安装git（安装包上传在本仓库）
2. 打开git bash，设置登陆账号，设置代理

git config –global user.name ‘’

git config –global user.email ‘’

git config –global http.proxy ‘XXX’

clone项目

git clone https://github.com/sunasitA/skillMap.git D:\skillmap

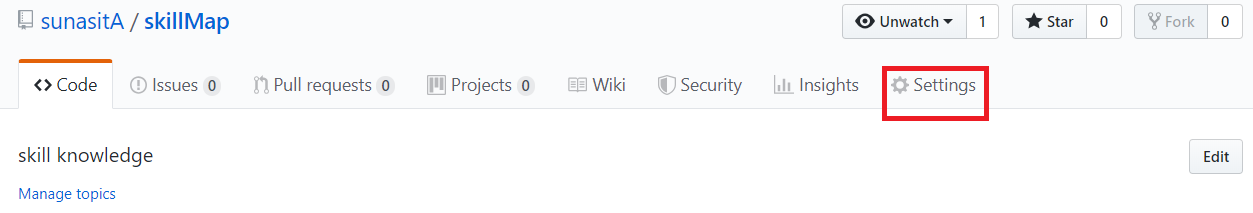
报错：fatal: HttpRequestException encountered.

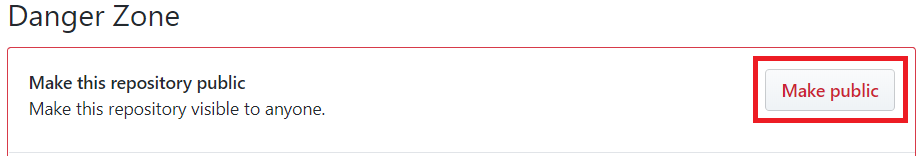
remote: Repository not found.

fatal: repository 'https://github.com/sunasitA/skillMap.git/' not found

原因：skillMap仓库是private的，修改为public即可

修改方法：





# 修改和提交本地文件到远程仓库

1. 在本地仓库新建多个文件
2. Git bash提交文件到远程仓库

git add .

git commit –m ‘第一次提交’

git push -u origin master

此处报错：

remote: Permission to sunasitA/skillMap.git denied to XXX.

原因电脑记了两个账号，要ssh到另一个，修改.git/cofig中的url

cd 到项目中，编辑 .git/cofig中的url

vim .git/config

将

[remote "origin"]

url = https://github.com/git的用户名/项目名称

的url改为

url = https://git的用户名@github.com/git的用户名/项目名称

即url = <https://sunasitA@github.com/sunasitA/skillMap.git>

再执行 git push -u origin master

输入账户，即可提交

1. 查看github网站，所有文件提交成功

# 撤销commit&push到远程仓库的提交

1. 撤销commit&push到远程仓库的提交
2. 修改文件，提交

git add .

git commit -m '这是次恶意提交'

git push origin master

1. 撤销本次提交

git log

|  |
| --- |
| commit e51bb4de709f321c72c4dc8ab348c18665247959 (HEAD -> master, origin/master, origin/HEAD)  Author: sunasitA <XXXX@XXXX.com>  Date: Fri Aug 23 17:25:55 2019 +0800  这是次恶意提交  commit c84f39c3f83db5692337a41a1a16bb07693aabfe  Author: sunasitA <XXXX@XXXX.com>  Date: Fri Aug 23 10:20:13 2019 +0800  删除临时文件 |

拿到要撤回到版本的ID c84f39c3f83db5692337a41a1a16bb07693aabfe

git reset --hard e51bb4de709f321c72c4dc8ab348c18665247959

git log

|  |
| --- |
| commit c84f39c3f83db5692337a41a1a16bb07693aabfe (HEAD -> master)  Author: sunasitA <XXXX@XXXX.com>  Date: Fri Aug 23 10:20:13 2019 +0800  删除临时文件 |

1. 强制提交当前版本号

git push origin master –force

# 找回git reset –hard的版本

1. 对于git reset –hard错误回退的代码，通过下面步骤找回。
2. git reflog 找到git reset –hard的版本号
3. git reset –hard (1)的id
4. 将文件拷贝出来
5. 替换本地工作区的文件
6. git add .

git commit –m ‘’

git push origin master

1. 什么是里程碑

完整可用版本的release。



1. 分支开发和合并到master

Mster分支推送了新的没有测试的代码，而已发布的代码中存在一个bug，此时就要在已发布代码上创建一个分支。

1. 创建一个新分支，切换到新分支

git branch newBranch

git branch –a

git checkout newBranch

1. 在新分支上进行修改

直接在本地工作区修改文件，就是在当前分支开发

1. 提交

git add .

git commit

git push origin newBranch

1. 发布修改了bug的版本
2. 切换到master分支，查看4的修改没有反应到master分支
3. 合并现在的代码到master的分支

git merge newBranch

git status

git push origin master

git log

newBranch的修改已经merge到master分支

结束

# GIT暂存区

git log –stat 看到提交的文件变更统计

|  |
| --- |
| commit b7ac0a8761703b6b262f9dad5b4861f1a5955b8b (HEAD)  Author: sunasitA <sunnana.fnst@cn.fujitsu.com>  Date: Wed Nov 13 18:14:59 2019 +0800  冲突解决后提交  "vue/\347\254\224\350\256\260.docx" | Bin 253785 -> 253706 bytes  1 file changed, 0 insertions(+), 0 deletions(-) |

git log –pretty=oneline 日志在一行

|  |
| --- |
| sunnana.fnst@G08FNSTD160024 MINGW64 /d/skillmap (master|REBASE 1/2)  $ git log --pretty=oneline  8908d0c0ba2a67a2ee6c64a760b3d26c5d4e8577 (HEAD) test2  b7ac0a8761703b6b262f9dad5b4861f1a5955b8b 冲突解决后提交  a7fe513a1b943efd8d34be8fa8dfe3e4535648d8 vue笔记修改  f26b57d5851d98419794eb61ad62668df0564ce3 vue demo的提交：左右布局+路由+父子组件间传值+兄弟组件间传值  6acc0baf3bc43d20926072f8c9a1f9f56ff22b15 解决冲突 |

git status –s 精简形式输出

|  |
| --- |
| sunnana.fnst@G08FNSTD160024 MINGW64 /d/skillmap (master|REBASE 1/2)  $ git status -s  A "git\345\256\211\350\243\205\345\222\214\344\275\277\347\224\250/~$\345\255\246\344\271\240\347\254\224\350\256\260.docx"  MM "git\345\256\211\350\243\205\345\222\214\344\275\277\347\224\250/\345\255\246\344\271\240\347\254\224\350\256\260.docx" |

第一列M 版本库和缓存区的差异

第二列M 工作区和缓存区的差异

git diff 显示工作区和缓存区的差异

git diff HEAD 显示工作区和版本库的差异

git diff –cached 显示缓存区和版本库的差异

缓存区

工作区

版本库

git diff HEAD

git diff

git diff --cached

将最近一次git commit提交到缓存区的内容撤销

git reset [filename]

git reset HEAD [filename]

将暂存区回退到上一次的提交前

git reset --hard HEAD^