

## § Documentation §

### Section 1: GitHub

GitHub, 是一个面向开源的软件存储和管理平台, 因为只支持 Git 作为唯一的版本库格式进行托管, 故名为 GitHub。由于可以轻松的寻找到大量的开源代码文件, GitHub 已经成为了管理软件开发等功能的首选方法, 被誉为“程序员的维基百科”。但 GitHub 使用 git 分布式版本控制系统, 是基于 Linux 系统开发而创造的, 对 Windows 系统使用者来讲并不算友好, 因此需要系统的学习才能熟练应用。

#### 1. 前期准备

GitHub 的官方网址为<https://github.com><sup>1</sup>。进入网页后点击 sign up 进行注册, 按照页面信息提示填写注册表单创建账号<sup>2</sup>。选择免费公开的仓库 (Unlimited public repositories for free), 跳过用户调查网站, 即注册成功, 并进入 GitHub 主页。

进入 GitHub 自己的主页后<sup>3</sup>, 需要先新建仓库和建立站点。点击仓库 (Repositories)→ 新建 (New), 进入新建仓库 (New repository) 页面。完成仓库的命名, 并简要进行描述 (Description), 选择公开 (Public) 并勾选添加 README file, 最后点击创建仓库 (Create repository) 即可完成。

图 1-1: 创建仓库页面

成功创建仓库后, 可以点击 Repositories 查看所有已建立的仓库。点击仓库名可以进入该库页面, 了解其基本信息。点击进入设置 (Settings) 页面, 具有修改仓库名字、修改默认分支 (branch) 名字等功能。设置中同样可以删除仓库, 将页面拉到最下方, 可以看到危险区 (Danger Zone), 点击最下方的删除此库

<sup>1</sup> 国内进入此网站不太稳定, 推荐使用 VPN 连接

<sup>2</sup> 使用 Google 邮箱账号注册更为方便

<sup>3</sup> 例:<https://github.com/lxlAppleMuscle>

(Delete this repository) 进入删除页面。进行删除操作时,不仅要多次确认,还要手动输入项目的名称以防止误删,最后还要输入 GitHub 的密码进行验证,才能成功删除仓库。

## 2. 下载与上传文件

完成仓库的创建后,就可以进行文件的下载和上传;也可以进入他人的 GitHub 主页,下载优秀的代码和资料,极大的方便了知识的传播与分享。但通过页面下载和上传方式较为繁复,为了更加轻松方便的进行资源的上传与下载,我们将学习 Git 分布式版本管理工具的使用。

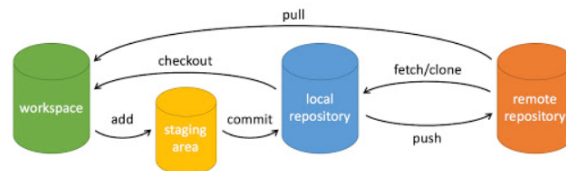


图 1-2: Git 基本操作示意图

### (1) Git 的安装与配置

Git 软件的官网为<https://git-scm.com/downloads><sup>1</sup>。安装下载过程并不复杂,按照流程一步一步进行即可,配置选择默认即可,注意安装路径为纯英文路径,防止报错。安装完成后,点击鼠标右键即可打开 Git Bash 页面:

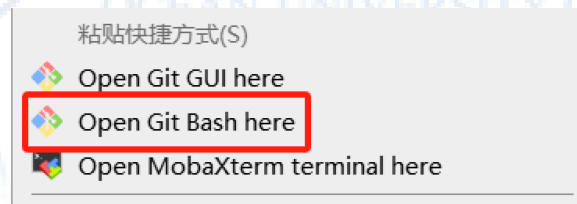


图 1-3: 打开 Git Bash

Git 客户端安装后,需要使用 SSH 与 GitHub 远程连接:

```
git config --global "UserName" #连接注册名
git config --global "UserMail" #连接注册邮箱
ssh-keygen -t rsa -C "UserMail" #生成SSH
```

完成上述工作后,会在 C:/User/用户/.ssh 路径下存放 SSH 文件,其中 id\_rsa 为私钥, id\_rsa.pub 为公钥。复制 id\_rsa.pub 中全部内容,打开 GitHub→个人账户 →Setting, 选择 SSH and GPG keys, 点击 New SSH key, 自定义 Title 然后粘贴上文复制的内容,完成连接。

```
ssh -T git@github.com
```

<sup>1</sup>官网速度较慢,可以在腾讯软件中心下载,[https://pc.qq.com/detail/13/detail\\_22693.html](https://pc.qq.com/detail/13/detail_22693.html)

```
lixianlong@ddd-Apple-muscle MINGW64 /d/profe
$ ssh -T git@github.com
Hi lxlApplemuscle! You've successfully authe
e shell access.
```

图 1-4: 成功完成连接展示

## (2) 推送文件

- 1 已经建立远程仓库之后,在本地建立一个文件夹,在命令行中打开该文件夹并执行 `git init` 命令,将该文件夹初始化成一个仓库。
- 2 在执行 `clone` 命令后,可以将远程仓库中的内容 clone 到本地仓库中,实现文件的下载:

```
git clone git@github.com:UserName/RepoName.git
```

```
lixianlong@ddd-Apple-muscle MINGW64 /d/professional c
it (master)
$ git clone git@github.com:lxlApplemuscle/Manual.git
Cloning into 'Manual'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-r
Receiving objects: 100% (3/3), done.
```

图 1-5: clone 命令完成效果

- 3 向本地仓库中手动添加拟推送远程仓库的文件及文件夹<sup>1</sup>,用 `ls` 可以看到本地仓库中的文件,然后在本地仓库中执行 `add` 命令,添加文件到暂存区。

```
git add signalFile    #添加单个文件
git add File1/ File2/ #添加多个文件夹,文件夹之间用空格隔开
git add .              #直接添加当前仓库下所有文件
git status             #查看仓库当前的状态,显示有变更的文件
```

- 4 文件进入暂存区后,执行 `commit` 命令提交到本地仓库,最后执行 `push` 命令,将文件推送至远程仓库中,完成文件的上传。

```
git commit -m "注释" #-m用于提交暂存区的文件,注释可以简单记录信息
git push origin branchName #提交文件于目标分支,默认为main分支
```

<sup>1</sup>注意:此处手动向本地仓库中添加文件并不意味着已上传至本地仓库中(可以看作你想存储的“货物”已经进了清单,但还没有入库)

## Section 2: Linux



中国海洋大学  
OCEAN UNIVERSITY OF CHINA



未来海洋学院  
Academy of Future Ocean