

运用GitLab客户端进行项目协作和管理

Git 是什么，它能干什么

这里我引用别人的一篇文章里的介绍，想了解更多 git 的使用可访问原文网址：

<https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/001373962845513aefd77a99f4145f0a2c7a7ca057e7570000>

Git 是什么？

Git 是目前世界上最先进的分布式版本控制系统（没有之一）。

Git 有什么特点？简单来说就是：高端大气上档次！

那什么是版本控制系统？

如果你用 Microsoft Word 写过长篇大论，那你一定有这样的经历：

想删除一个段落，又怕将来想恢复找不回来怎么办？有办法，先把当前文件“另存为……”一个新的 Word 文件，再接着改，改到一定程度，再“另存为……”一个新文件，这样一直改下去，最后你的 Word 文档变成了这样：



过了一周，你想找回被删除的文字，但是已经记不清删除前保存在哪个文件里了，只好一个一个文件去找，真麻烦。

看着一堆乱七八糟的文件，想保留最新的一个，然后把其他的删掉，又怕哪

天会用上，还不敢删，真郁闷。

更要命的是，有些部分需要你的财务同事帮助填写，于是你把文件 Copy 到 U 盘里给她（也可能通过 Email 发送一份给她），然后，你继续修改 Word 文件。一天后，同事再把 Word 文件传给你，此时，你必须想想，发给她之后到你收到她的文件期间，你作了哪些改动，得把你的改动和她的部分合并，真困难。

于是你想，如果有一个软件，不但能自动帮我记录每次文件的改动，还可以让同事协作编辑，这样就不用自己管理一堆类似的文件了，也不需要把文件传来传去。如果想查看某次改动，只需要在软件里瞄一眼就可以，岂不是很方便？

这个软件用起来就应该像这个样子，能记录每次文件的改动：

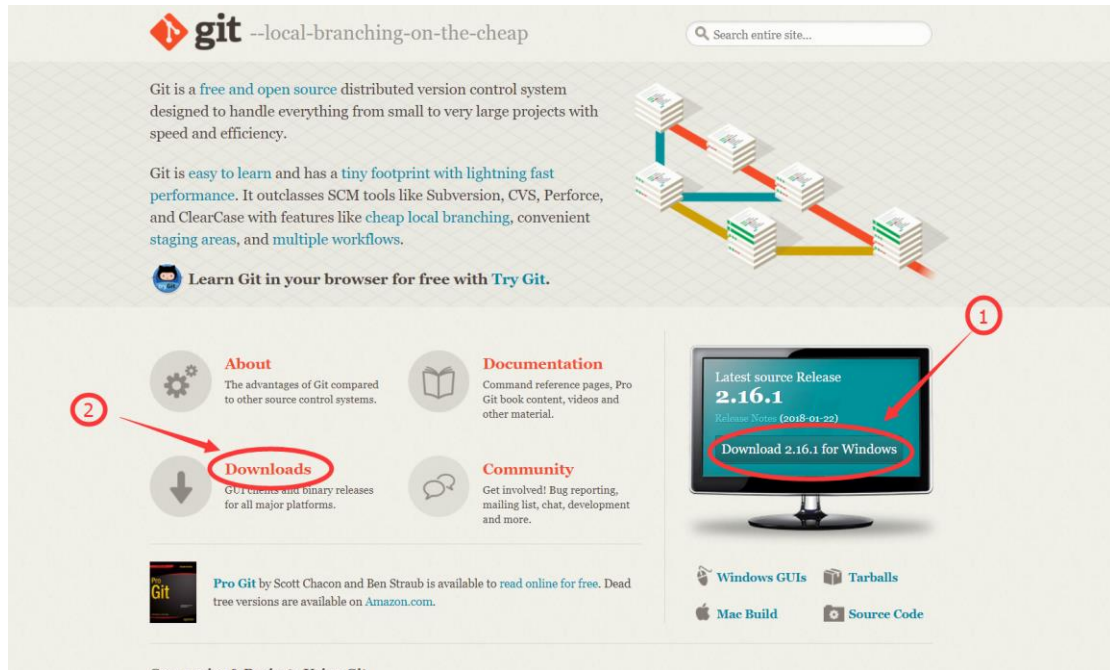
版本	文件名	用户	说明	日期
1	service.doc	张三	删除了软件服务条款5	7/12 10:38
2	service.doc	张三	增加了License人数限制	7/12 18:09
3	service.doc	李四	财务部门调整了合同金额	7/13 9:51
4	service.doc	张三	延长了免费升级周期	7/14 15:17

这样，你就结束了手动管理多个“版本”的史前时代，进入到版本控制的 21 世纪。

需要安装的软件

Git

首先我们需要去官网下载 git 地址：<https://git-scm.com/> 下载完成后，我们就可以安装 git, 安装过程点下一步就行，唯一需要注意的就是下载适合你系统的版本，windows 版本可以点击①处，其他版本点击②处。这里我的 pc 为 windows, 所以下载 windows 版本，如下图：



GitHub 客户端

GitHub 客户端可以简化我们对 git 的使用，帮助你快速上手 git，下载方式：可以登陆我们的本地 git 服务器，点击适合你的版本如下图所示 ① 处,当然你也可以去官网下载不同的版本，点击 ② 处。



4DAGE-Git



开始工作

接着就进入正题，运用 github 客户端在我们的 git 服务器上项目进行协作管理。

创建账号并登陆

填写基本信息, 然后记住你的**账号**和**密码**(后面在使用 github 客户端的时候也会用到), 用你创建的账号和密码登陆

注册

用户名 *

邮箱 *

密码 *

确认密码 *

766973

验证码 *

创建帐户

已经注册? 立即登录!

登录

用户名或邮箱 *

密码 *

☐ 记住登录

登录 忘记密码?

还没帐户? 马上注册。

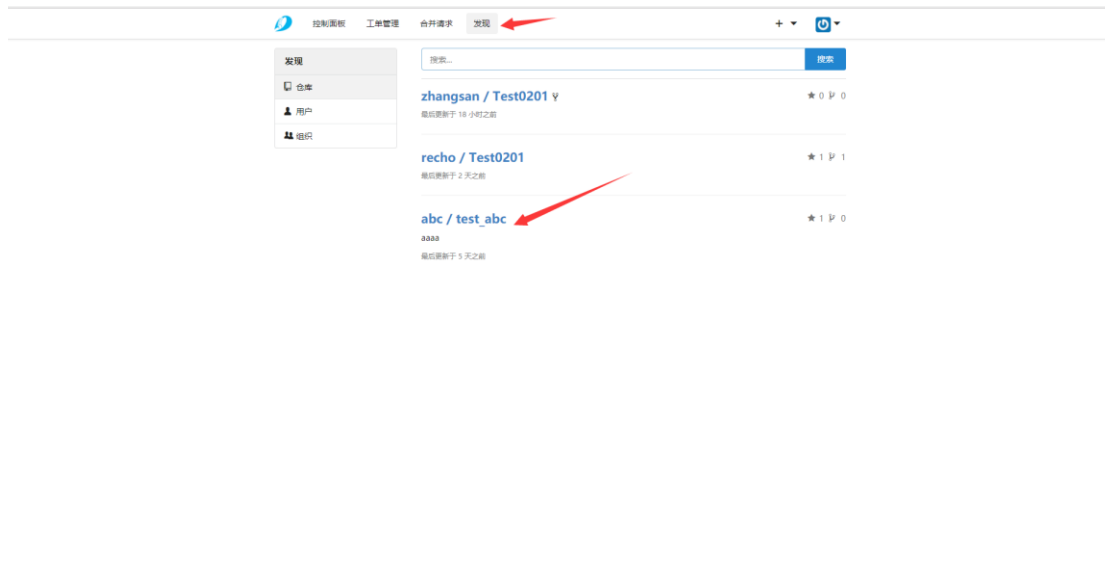
克隆服务器上的仓库

要想 clone 远程服务器上的仓库，首先，在你的远程服务器上找到你仓库的地址(没有仓库的可以点击**发现**，查看服务器上的**公开仓库** (不能发现私有仓库，后面会介绍私有仓库

的创建)，当然你也可以自己**创建**一个仓库，然后 clone)，然后可以通过 http 或者 ssh 协议将远程服务器上的仓库克隆到本地。

第一步：发现别人的共有仓库和创建仓库

点击**发现**，查看服务器上的公有仓库



点击**控制面板**，然后点击**+号**创建自己的仓库



填写基本信息，创建仓库，这里说明一下如果不想让别人发现你的这个仓库，可以选中私有仓库选项，其他默认即可

The image shows the 'Create New Repository' form in the Gogs web interface. The form has the following fields and options:

- Owner:** A dropdown menu showing 'recho'.
- Repository Name:** A text input field containing 'temp'. Below it is a hint: '伟大的仓库名称一般都较短、令人深刻并且独一无二的。'
- Visibility:** A section with a red circle around it. It contains a checkbox labeled '该仓库为 私有的' (This repository is private), which is selected. A red arrow points to this checkbox.
- Repository Description:** A large text area for describing the repository.
- .gitignore:** A dropdown menu showing '选择 .gitignore 模板' (Select .gitignore template).
- 授权许可:** A dropdown menu showing '请选择授权许可文件' (Please select a license file).
- 自述文档:** A dropdown menu showing 'Default'.
- Use selected files and templates to initialize the repository:** A checkbox that is currently unchecked.
- Buttons:** A green '创建仓库' (Create Repository) button and a grey '取消' (Cancel) button.

第二步：打开 github 客户端将远程服务器仓库克隆到本地

复制仓库地址，地址有两种，http 方式和 ssh 方式，这里选择 http

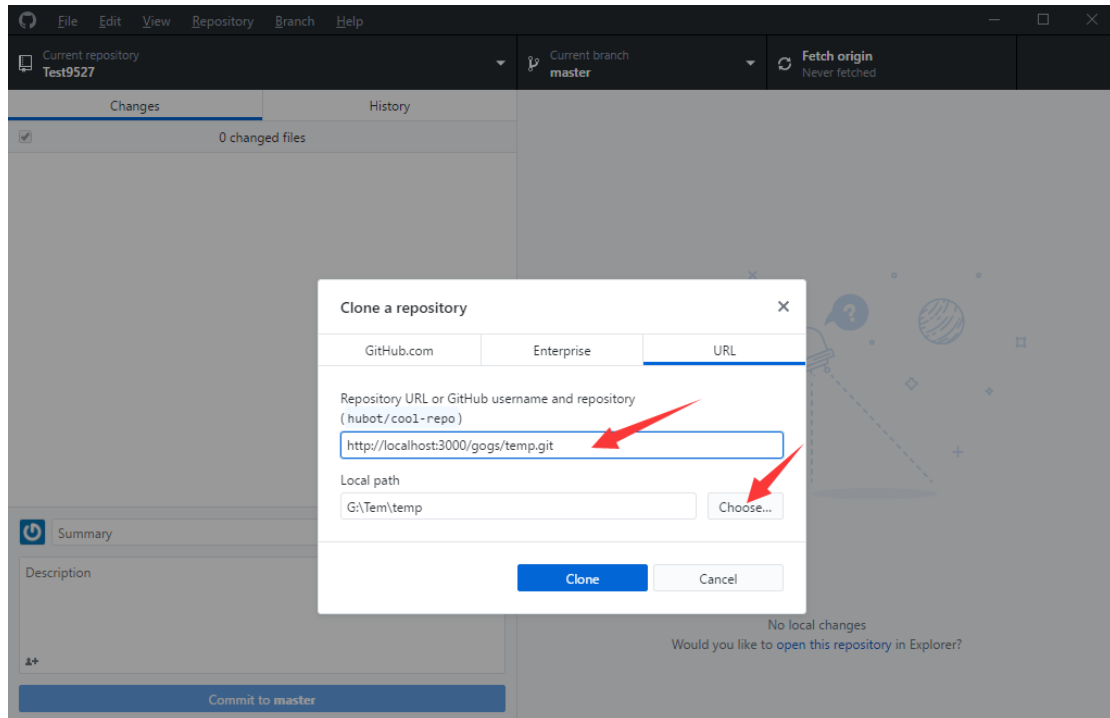
The screenshot shows the Gogs web interface for a repository named 'gogs / temp'. The top navigation bar includes links for '控制面板', '工单管理', '合并请求', and '发现'. The repository page shows '取消关注' (1), '点赞' (0), and tabs for '文件', '工单管理 (0)', and 'Wiki'. A '仓库设置' (Repository Settings) link is also present. The main content area is titled '快速帮助' (Quick Help) and contains the following sections:

- 克隆当前仓库** (Clone this repository): A section with a red arrow pointing to the 'HTTP' button and another red arrow pointing to the '复制' (Copy) button. The URL 'http://localhost:3000/gogs/temp.git' is displayed.
- 从命令行创建一个新的仓库** (Create a new repository from the command line): A code block containing the following commands:

```
touch README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin http://localhost:3000/gogs/temp.git
git push -u origin master
```
- 从命令行推送已经创建的仓库** (Push an already created repository from the command line): A code block containing the following commands:

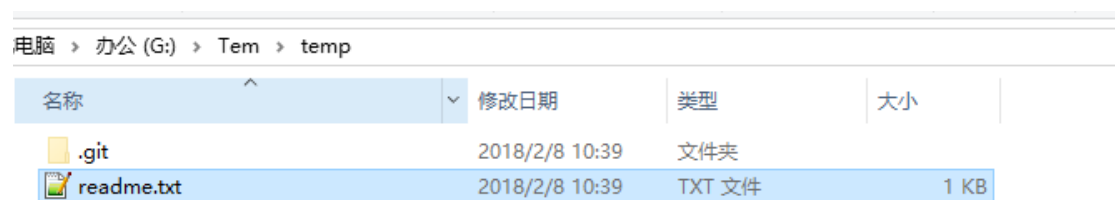
```
git remote add origin http://localhost:3000/gogs/temp.git
git push -u origin master
```


打开客户端，点击 File->Clone Repository..，clone 远程仓库，选择一个本地的保存路径

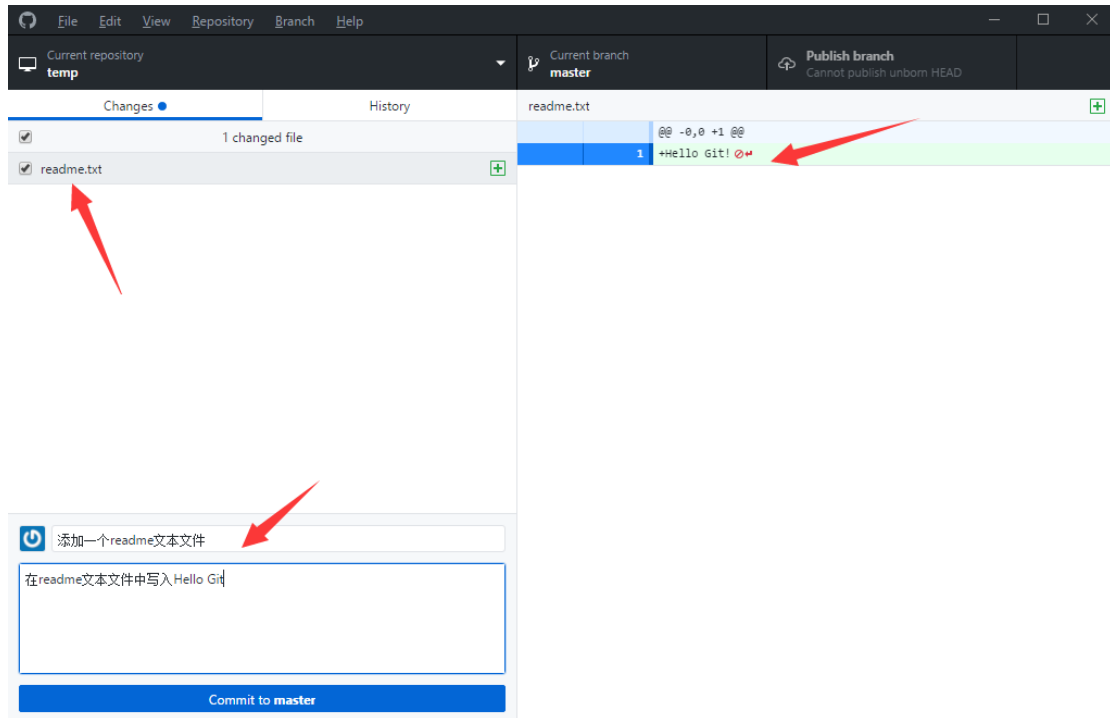


编辑本地仓库并推送本地仓库的更新

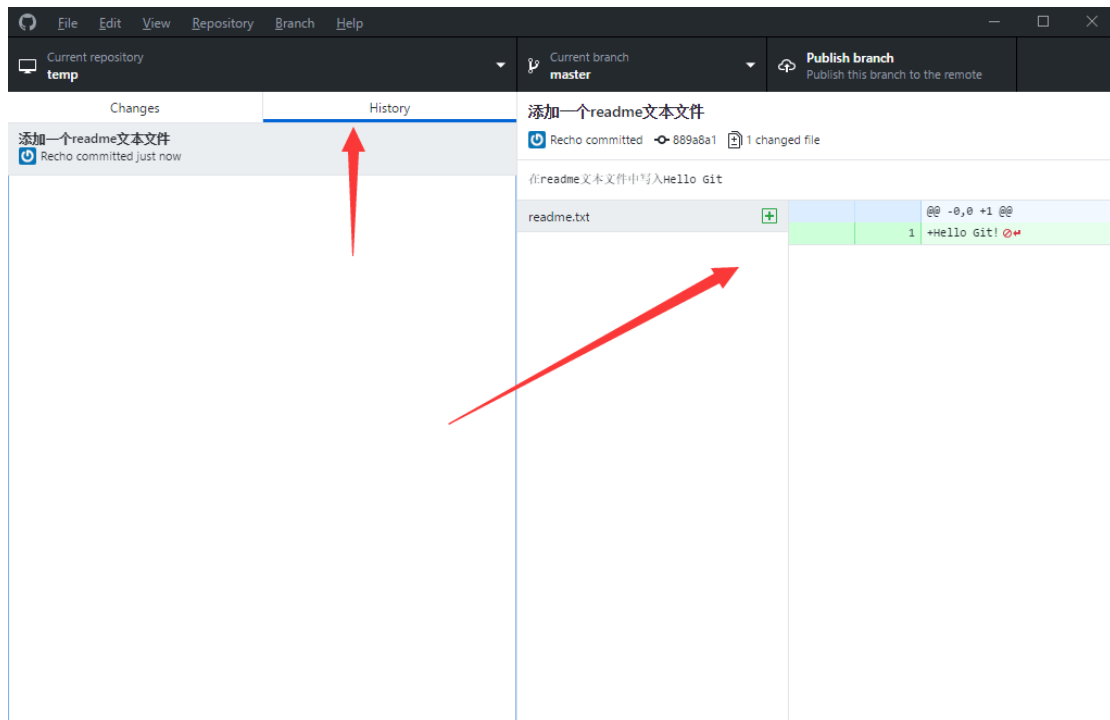
打开刚才保存的路径，然后添加一个文件，比如说 readme.txt 文本文件，写入 Hello Git



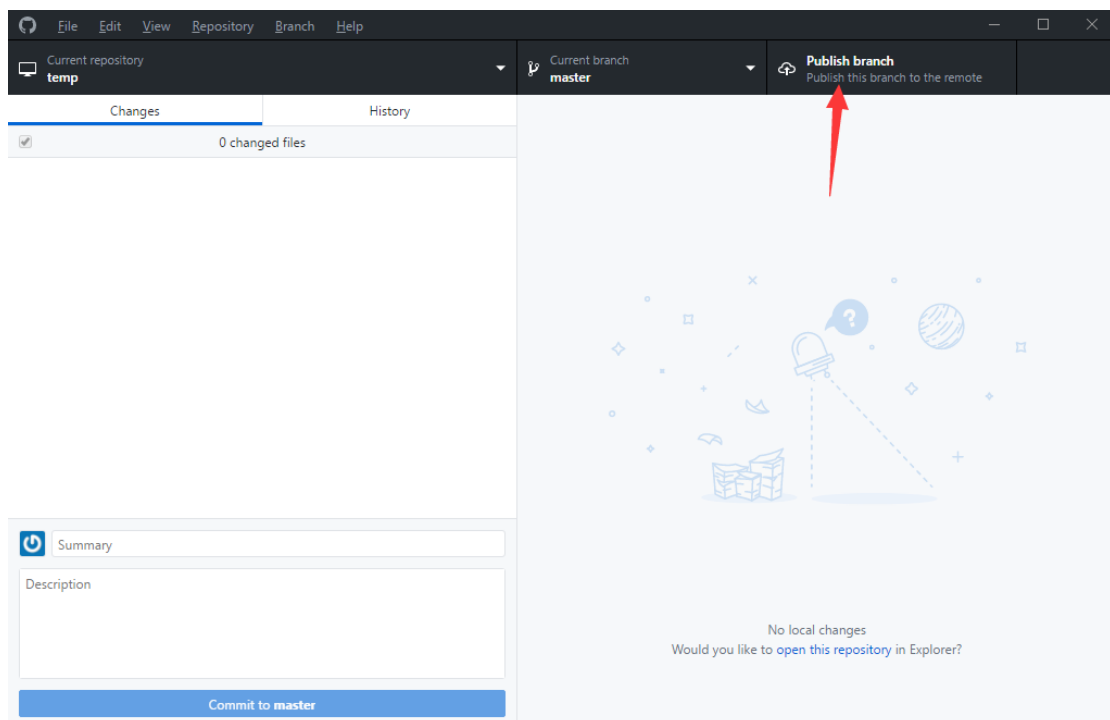
打开 github 客户端，查看更改，然后为本次更改添加一个提交，输入此次更改的主要目的和内容。



查看更改历史

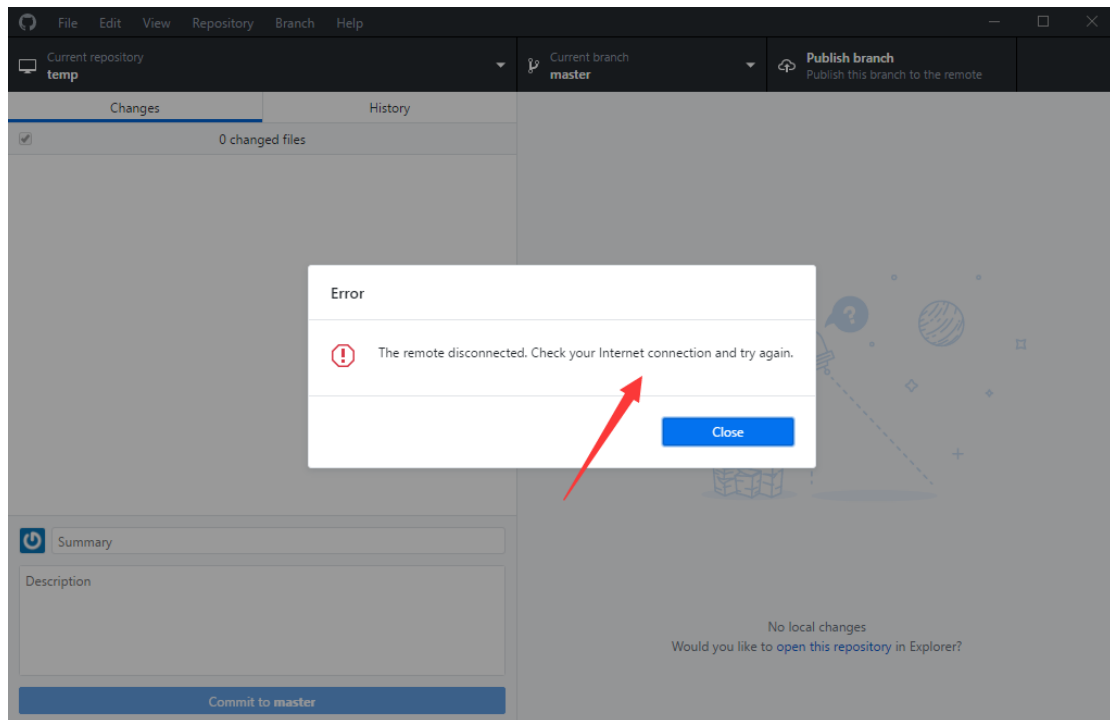


将本地更改推送到远程 git 服务器上



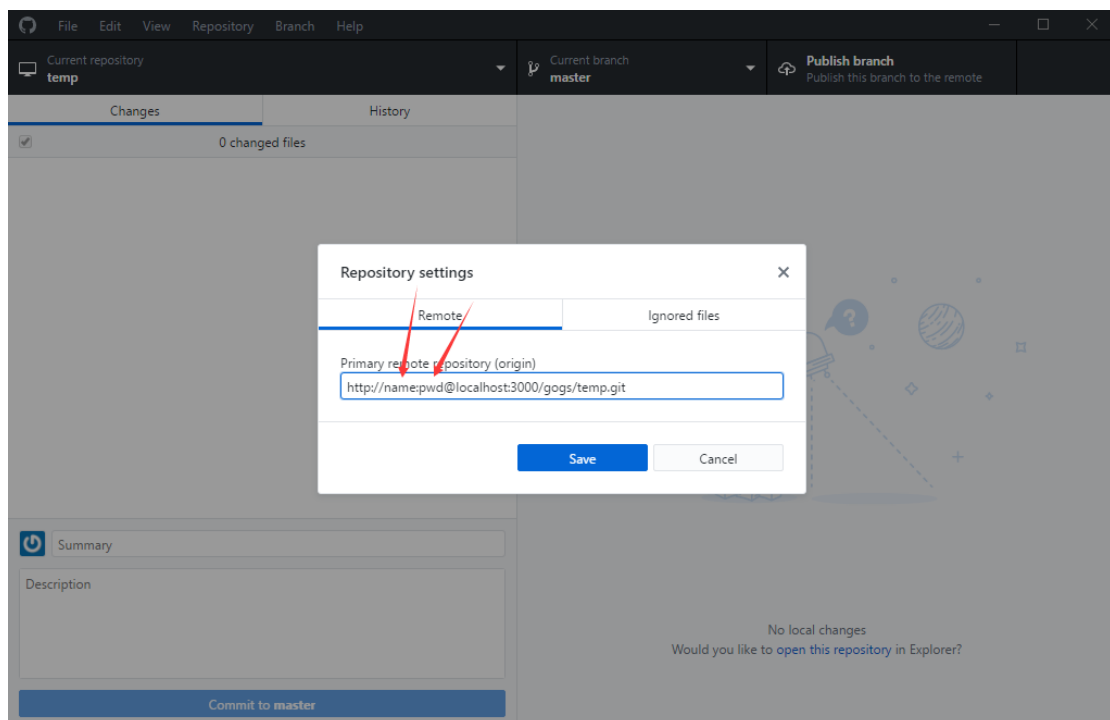
上面的操作可能出现的两种情况

第一种情况，会弹出让你输入账号和密码，这个就是你 git 服务器上的账号和密码
第二种情况，如下图所示，不能推送，主要是 windows 用户凭据引起的



解决办法：

点击 Repository->Repository Setting..更改仓库设置，加入用户名和密码，中间用冒号(:) 分开，最后要加一个@符号,设置完成后便可以点击推送了
原来的地址：http://localhost:3000/gogs/temp.git
更改后：http://用户名:密码@localhost:3000/gogs/temp.git



再来看看 git 服务器上的变化



SSH Clone 方式

大部分时候我们会选择 http 方式 clone 我们的项目，这样简单快速，但是任何事物都有两面性，比方说 ssh 的 clone 速度要快于 http,而且没有上面所说的凭据问题，下面说一下怎么通过 ssh 克隆项目吧

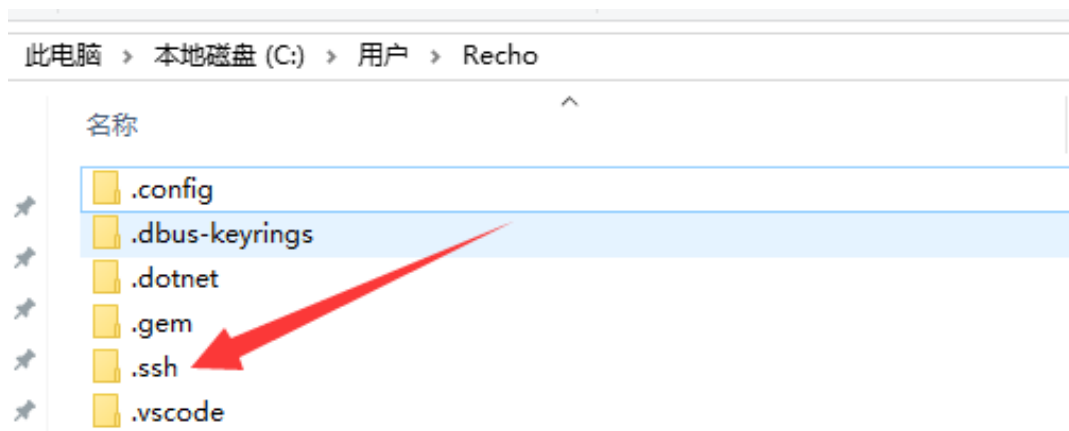
第一步：生成 ssh 密钥

要生成公钥，就要用到我们之前安装的 git 了，在桌面右键单击选择 **Git Bash Here** 打开 git 命令行，输入命令（需要注意 xxxxx@xxxxxx 为你当时在 git 服务器上设置的邮箱地址，在你的 git 远程用户设置里面可以查看，下面的 ssh 密钥设置也在哪里）：

```
ssh-keygen -t rsa -C "xxxxx@xxxxx.com"
```

然后一路回车键即可生成

打开 `~/.ssh` 查看你的 ssh 密钥（windows 的用户目录在 c 盘下的当前用户目录下）如下图



第二步：将公钥配置到你的远程仓库

打开第一步中提到的.ssh 目录，打开 id_rsa.pub，然后复制

名称	修改
 authorized_keys	201
 id_rsa	201
 id_rsa.pub	201
 known_hosts	201

将复制的内容粘贴

帐户设置

个人信息

头像设置

修改密码

邮箱地址

SSH 密钥

帐户安全

仓库列表

组织列表

授权应用

删除帐户

管理 SSH 密钥

增加 SSH 密钥

以下是与您帐户所关联的 SSH 密钥，如果您发现有陌生的密钥，请立即删除它！

需要帮助？请查看有关 [如何生成 SSH 密钥](#) 或 [常见 SSH 问题](#) 寻找答案。

密钥名称

pc

密钥内容

增加密钥

最后，我们就可以通过 ssh 来 clone 我们的项目了，复制 ssh 的 clone 地址，最后 clone 是与 http 一样，可以参照上面的 http 克隆方式。

recho / test1

取消关注 1 点赞 0

文件

工单管理

Wiki

仓库设置

快速帮助

克隆当前仓库 不知道如何操作？访问 [此处](#) 查看帮助！

HTTP

SSH

Recho@192.168.0.141:recho/test1.git

从命令行创建一个新的仓库

```
touch README.md
git init
git add README.md
git commit -m "first commit"
```

好了，本次的使用说明就是这些，如果有什么疑问可参照 github
官方帮助文档 <https://help.github.com/> 或者发 email 给我：
springworder@gmail.com