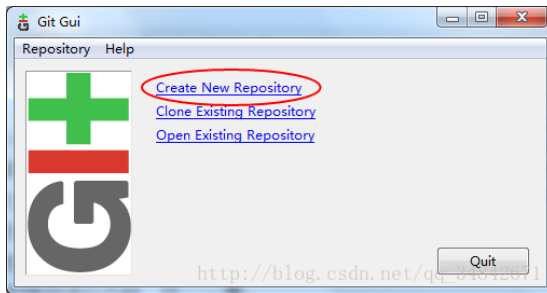


# Git 可视化——Git GUI 的使用

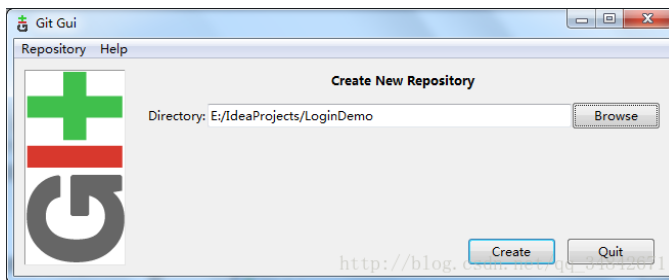
如果想简单地使用一下 Git 进行版本控制和团队开发，我们不妨试一试使用 Git GUI 的方式对 Git 进行操作。

## 1.在本地新建仓库

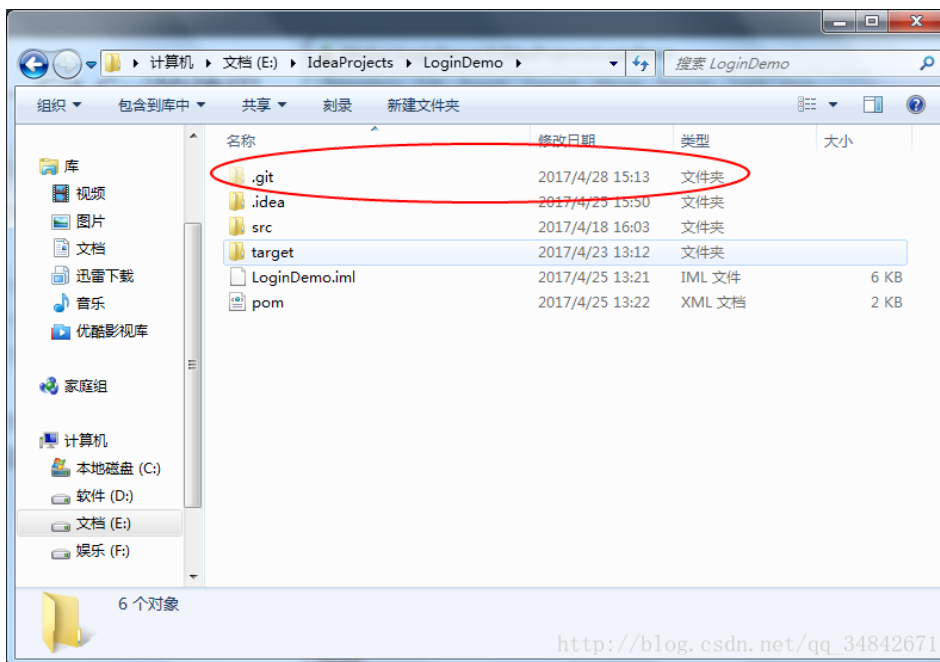
首先，我们打开 Git GUI 是这样的一个界面，选择第一项，新建仓库。



然后选择你需要进行版本管理的项目路径，我选择了一个 LoginDemo 的项目。

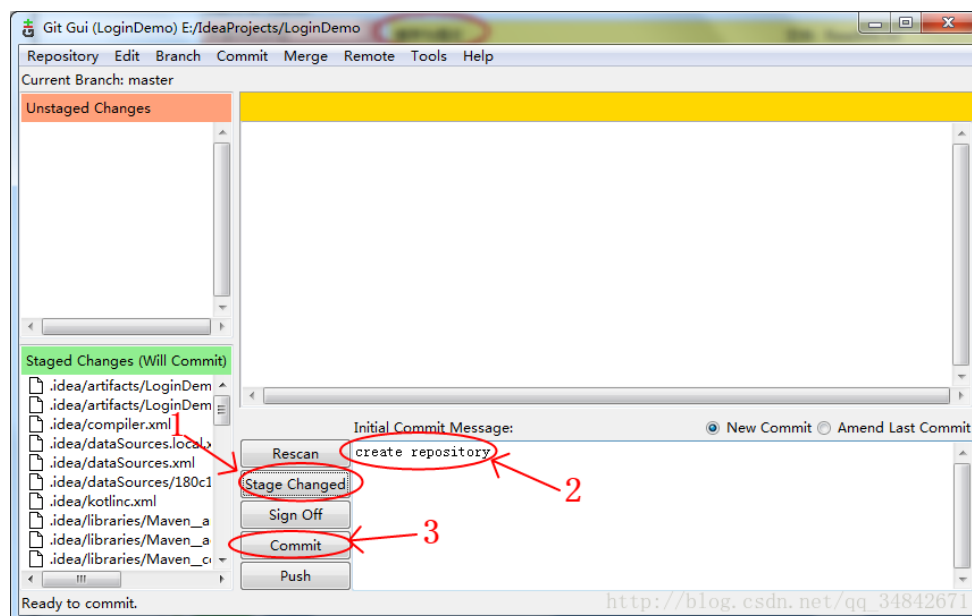


当你创建了仓库的时候，你可以在该项目的路径下看见多了一个.git 文件夹（设置了隐藏文件不可见的话看不见）

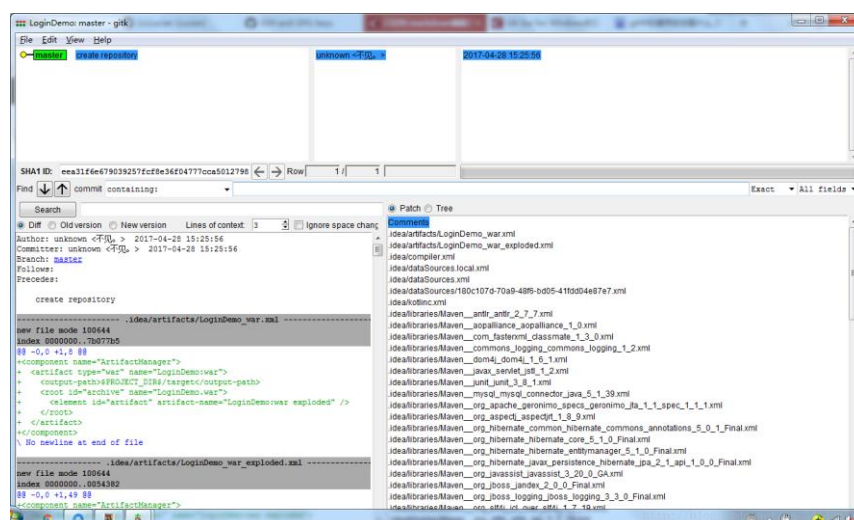


在 Git GUI 中，如果 Unstaged Changes(未缓存的改动)中包含文件，则先点击 Stage Changed，

将未缓存的改动加入缓存，再点击 Commit；否则需要先点击 Rescan 扫描项目中进行过改动的文件。注：提交描述必须要填写，不然无法提交。



提交成功后，我们可以在 Repository->Visualize All Branches History 中看到我们提交的历史记录（这是提交到本地而不是提交到远程服务器）。至此，本地仓库就创建成功了。



## 2.将项目提交到远程

上一步已经使用 Git 在本地建立起了仓库，然后我们需要将该项目提交到远程服务器以便同事或其他合作者共同参与开发。一般的开源项目使用 github 作为远程服务器。其实在本地简历仓库后，只需要点击 push 就可以直接上传。但是我们还没有将本地的 Git 与 github 建立联系以及设置安全协议。

首先，我们需要在有一个自己的 github 帐户，然后在 github 上新建一个 repository，名字也叫做 LoginDemo（可以和本地项目名不一样）。

## Create a new repository

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

Owner

Repository name

LLLLucian

 / 

LoginDemo

Great repository names are short and memorable. Need inspiration? How about **automatic-garbanzo**.

Description (optional)

LoginDemo about git test.

☒ Public

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

☐ Private

You choose who can see and commit to this repository.

☐ Initialize this repository with a README

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

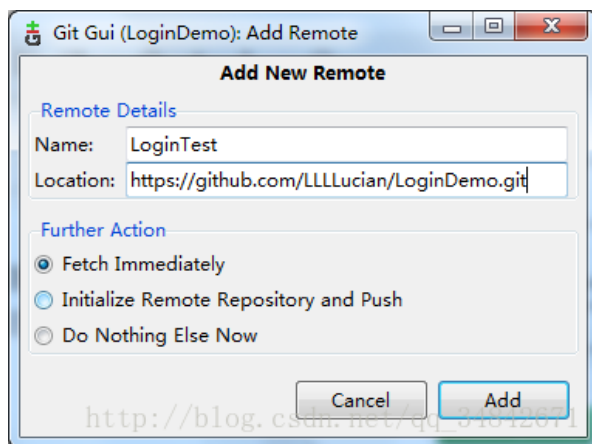
Create repository

[http://blog.csdn.net/qq\\_34842671](http://blog.csdn.net/qq_34842671)

在 Git Gui 中，选择 Remote->add 添加远程服务器，远程服务器信息有两种填写方式，填写 https 地址或 ssh 地址，对应 github 项目的 https 和 ssh 地址，推荐使用 ssh 方式。

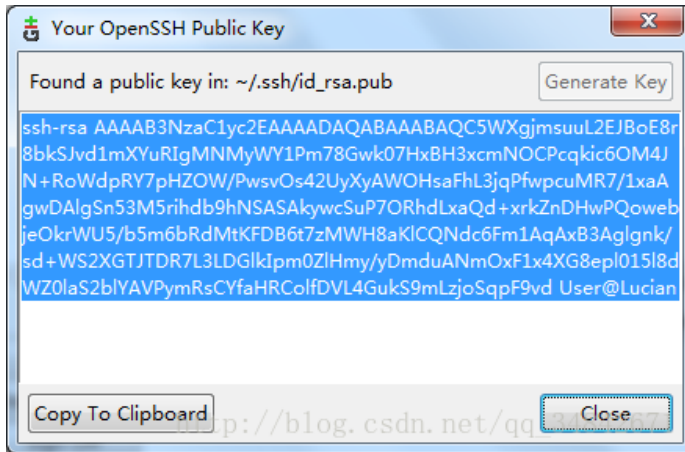
### https 地址

这种方式需要输入你的 github 帐户和密码，意味着通过这种方式你只能够操作自己的项目。

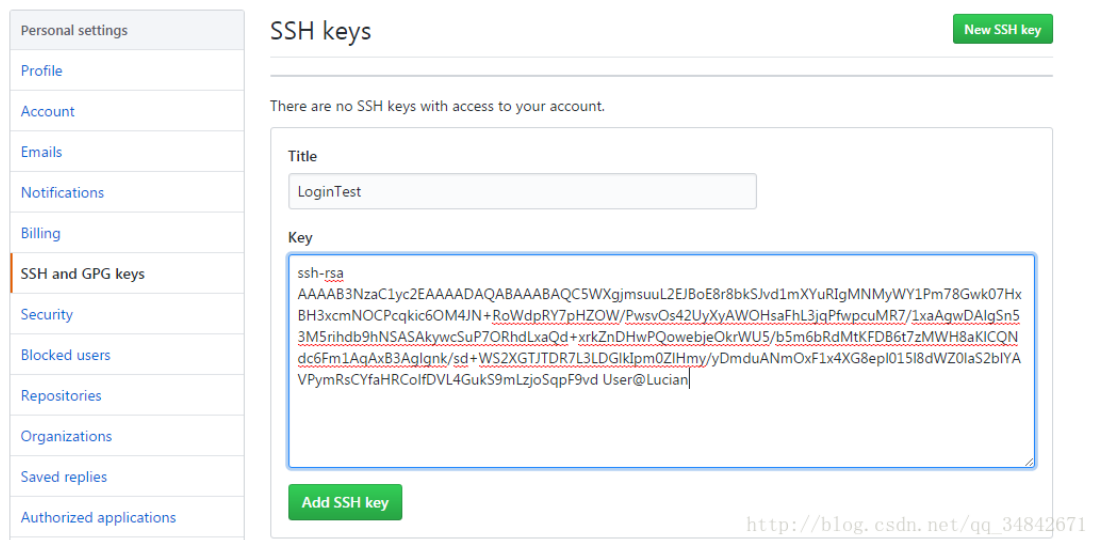


### ssh 地址

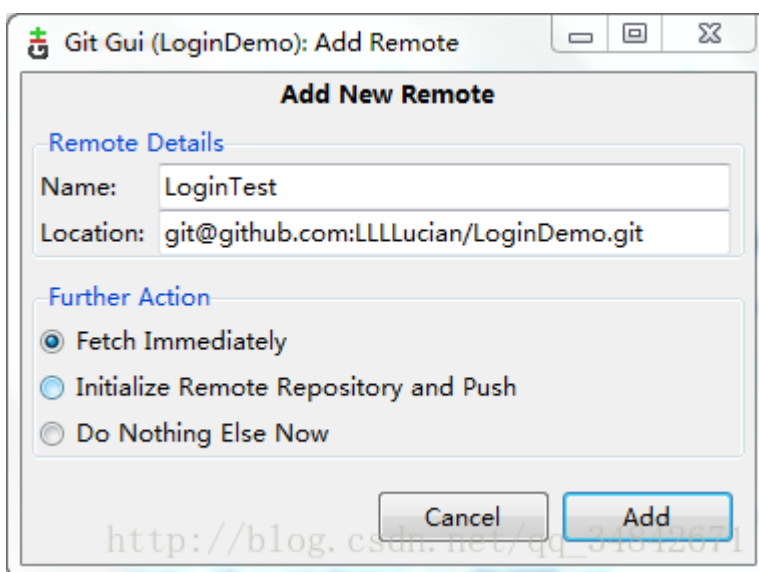
这种方式需要进行授权设置，在 Git Gui 的菜单栏，点击 Help->Show SSH key->Generate SSH KEY 创建密钥。



然后在 github 的 Personal settings 中添加它，title 随意，可以用 Home，company 等作为标识来区别。

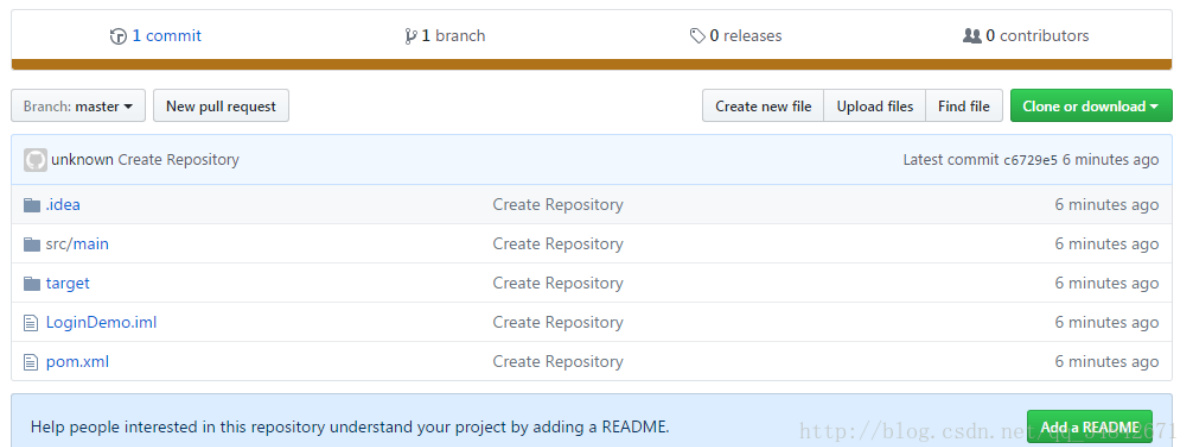


添加远程服务器信息。



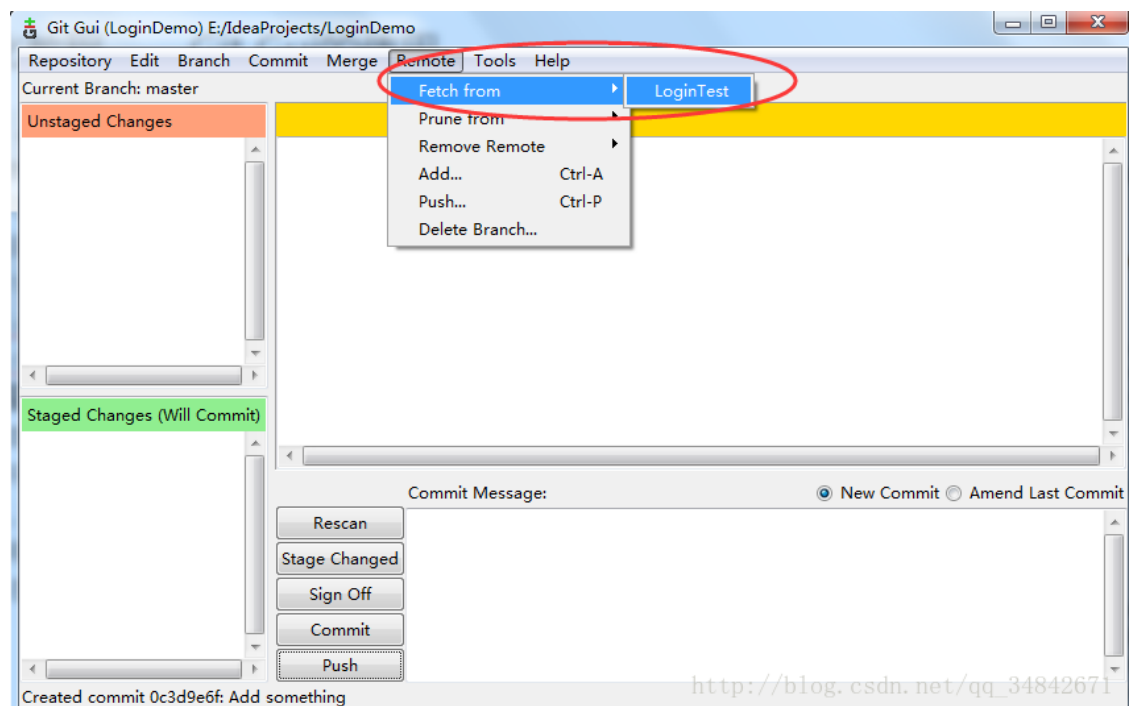
接下来，我们便可以直接在 Git Gui 点击 push 提交至远程客户端，刷新一下 github，便

可以看到项目已经在 repository 中了。



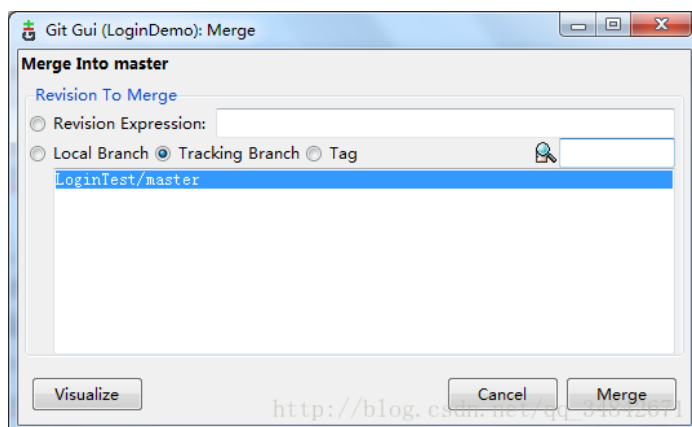
### 3.从远程下载更新

设想多人参与项目开发，每个人都只 push 到远程，完全不顾其他成员的进度，这样的开发模式无疑是有问题的。我们最好的做法是，每次 push 到远程的时候，先从远程把目前进度 fetch 下来，在和自己目前项目进度 merge 后，再将它 push 到远程。打开 Git Gui，在菜单栏中选择 Remote->Fetch from->LoginTest，便可以从远程服务器更新到本地，但是尚未与自己当前项目合并。

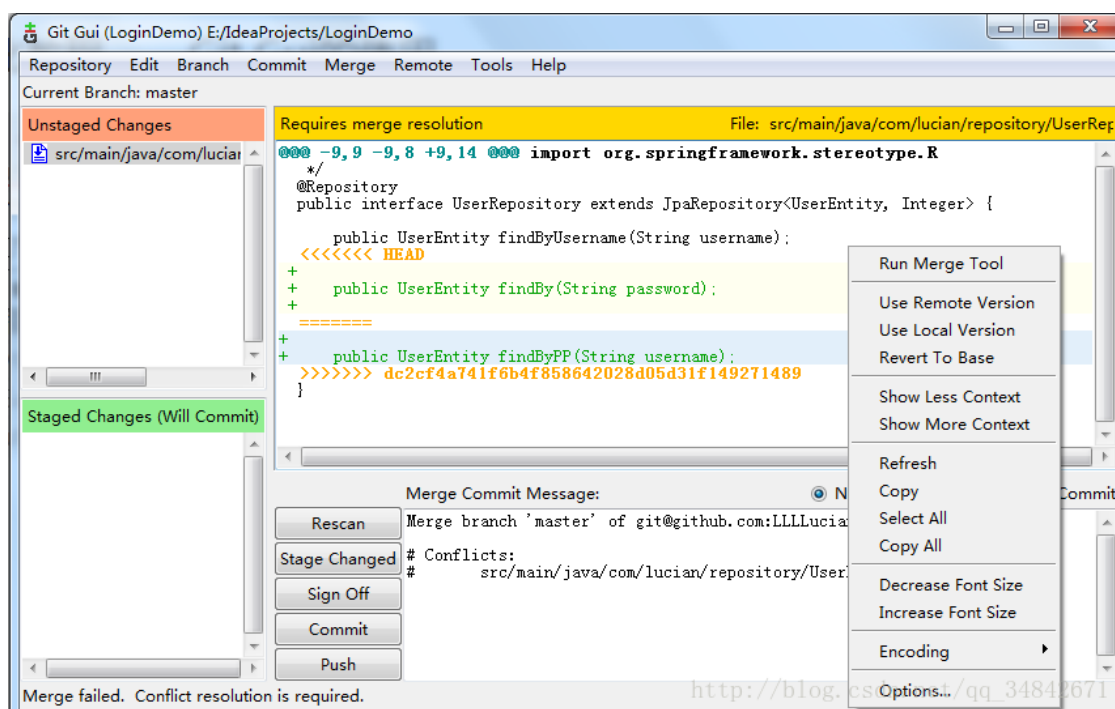


### 4.合并解决冲突

从远程 fetch 后，选择 Git Gui 的 Merge->Local Merge 进行合并，选择 Tracking Branch。

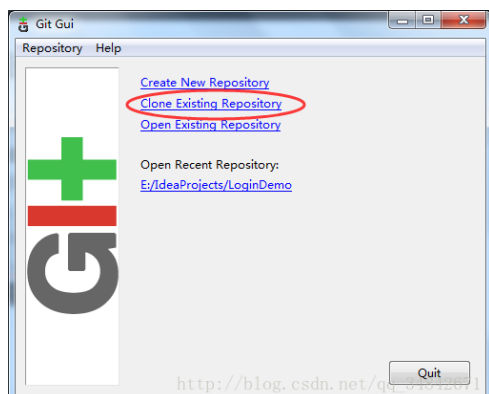


如果本地有一个方法名叫 `findUser`，而远程服务器中该方法的名字改变了，例如变成了 `findPassword`，则合并遇到冲突，可以右键空白部分选择保留本地版本或远程版本进行合并（图中绿色的代码表示冲突部分）。



## 5. 克隆 github 上的项目到本地

在 Git GUI 的主界面选择克隆已有仓库。



Source Location 即 github 中项目的地址，和第二部分（将项目提交到远程）中一样，可以选择 https 地址或 ssh 地址，Target Directory 是在本地存放该项目的路径。点击 Clone，成功从 github 将项目克隆到本地。

