# GitHub for Windows 使用手册

## 简介

GitHub的核心是由著名的Linux的团队开发的名为Git的开源版本控制系统(Version Control System. VCS)，大量的项目加入到GitHub证明使用Git为核心的GitHub关联到本地的文件进行版本控制是非常可靠的。

* *如果你还不了解什么是Git，请访问*

*http://git-scm.com/book/en/Getting-Started-Git-Basics*

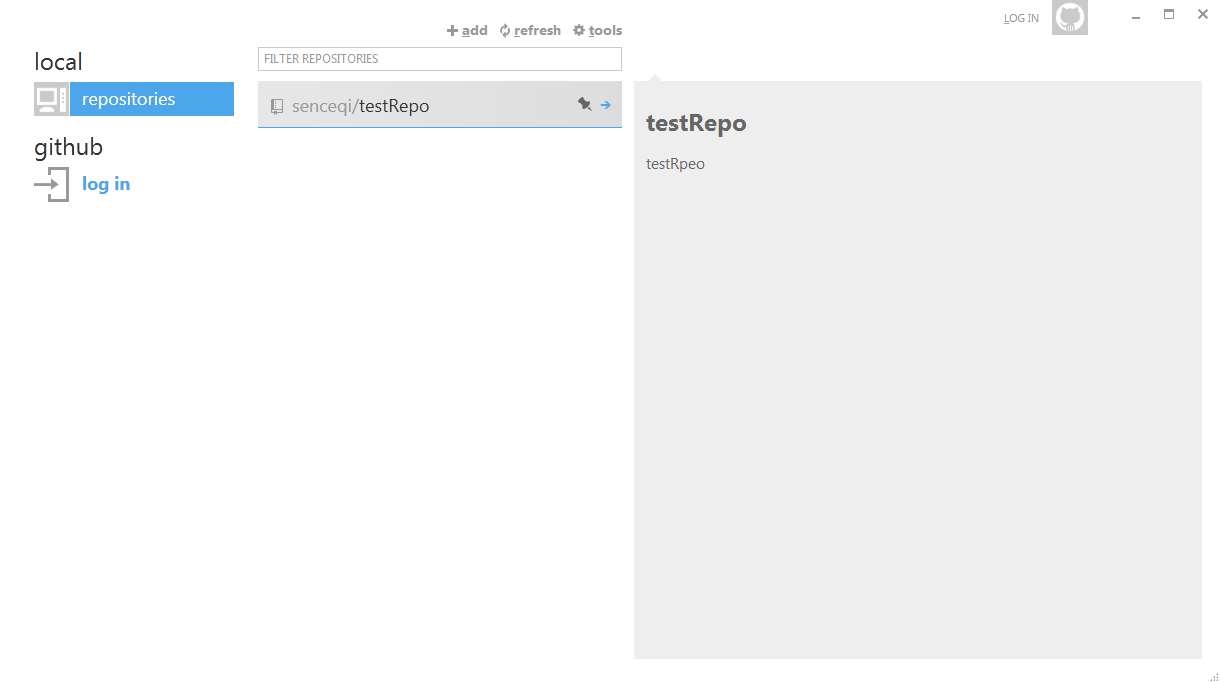
目前GitHub已经开始支持windows操作系统了，并开发出了一款名GitHub for Windows 的软件供我们方便的连接到GitHub服务器进行版本的控制管理，下面将介绍GitHub for Windows 的使用。

## 安装GitHub For Windows

首先需要下载GitHub For Windows，下载地址为：

*http://github-windows.s3.amazonaws.com/GitHubSetup.exe*

下载完毕后直接安装软件，需要半小时左右，软件自动安装完毕，安装完毕后，桌面上会有GitHub快捷图标，运行GitHub，界面将会如下图所示



**图1 GitHub安装后界面**

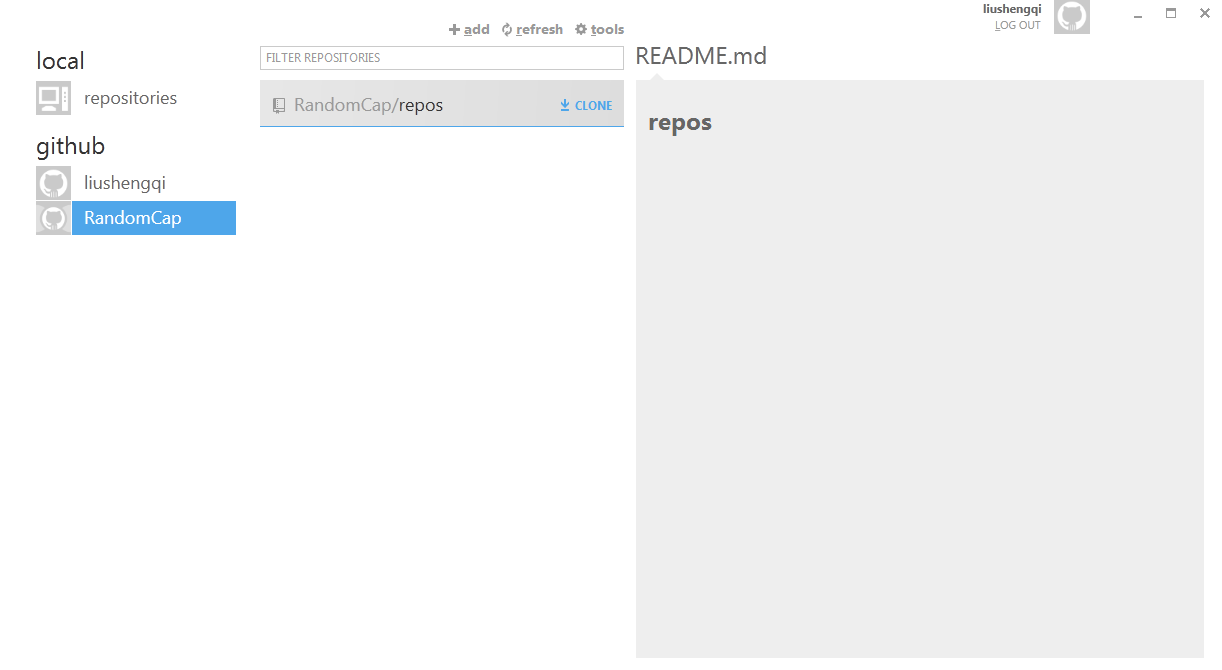
## 使用GitHub For Windows

使用GitHub For Windows之前我们需要在GitHub官方网站注册一个属于自己的帐号，注册地址为：*https://github.com/users*

* *注册完成后，貌似GitHub自己没有开放设置头像的功能，只能通过Gravatar*

*（http://en.gravatar.com/）来设置全球通用头像，但是Gravatar注册不了，所以还是设置不了头像*

注册完成后，运行安装的GitHub For Windows，点击右上角的LOG IN 进入登录界面，输入自己的帐号，下面以帐号“liushengqi”为例，登录成功后，界面将会如下图所示：



**图2 登录后界面**

如上图所示，左边的显示的分别是本地和GitHub服务器上的数据仓库(下面据图介绍)，中间显示的是点击左边具体栏时具体的数据仓库名，右侧则是选择中间的具体数据仓库名后该数据仓库的README.md文件内容。

* ***local栏****：显示的为本地存在的git数据仓库，一般是从服务器上clone下来的。*

***github栏****：显示的是登录帐号在GitHub服务器上的数据仓库，其中liushengqi是登录者帐号，RandomCap是liushengqi这个帐号所加入的组织(organization)，点击 liushengqi 或者 RandomCap界面中间显示的分别在GitHub服务器端属于liushengqi 和 RandomCap的数据仓库。*

### 3.1 从GitHub上Clone数据仓库

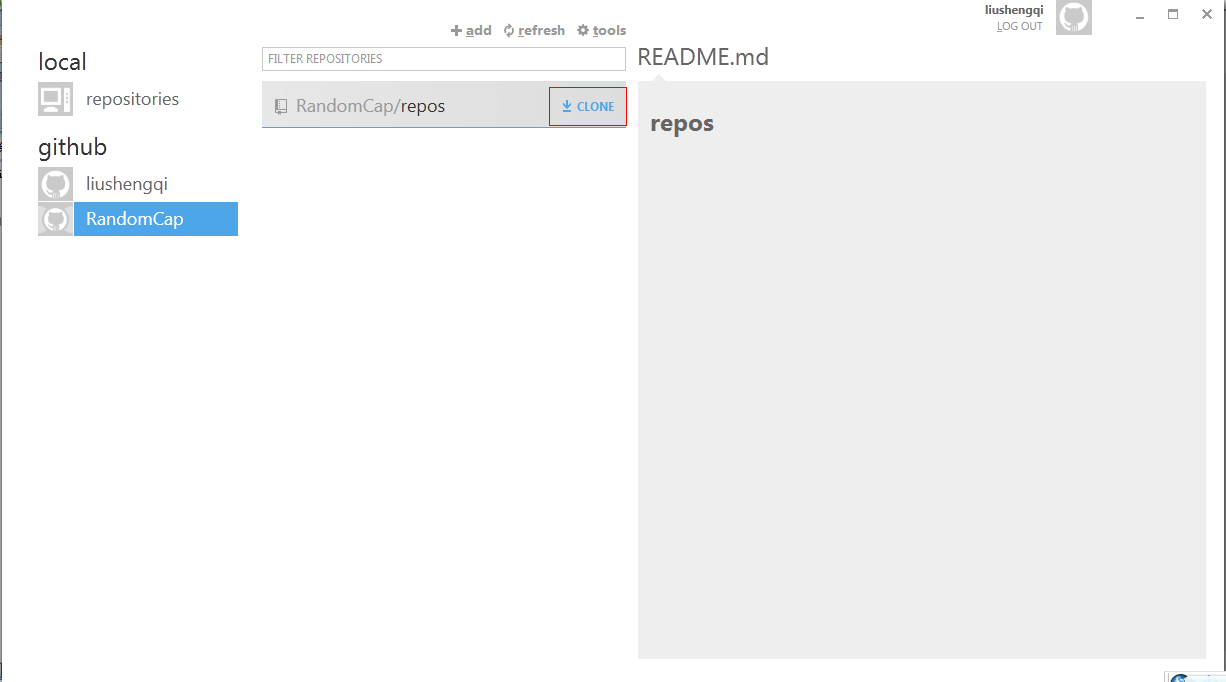
我们要合作团队开发一个项目，则首先需要从GitHub服务器上Clone下来需要合作的项目。

* *为什么不是fork下来，不了解fork？请参考*

*http://www.zhihu.com/question/20431718*

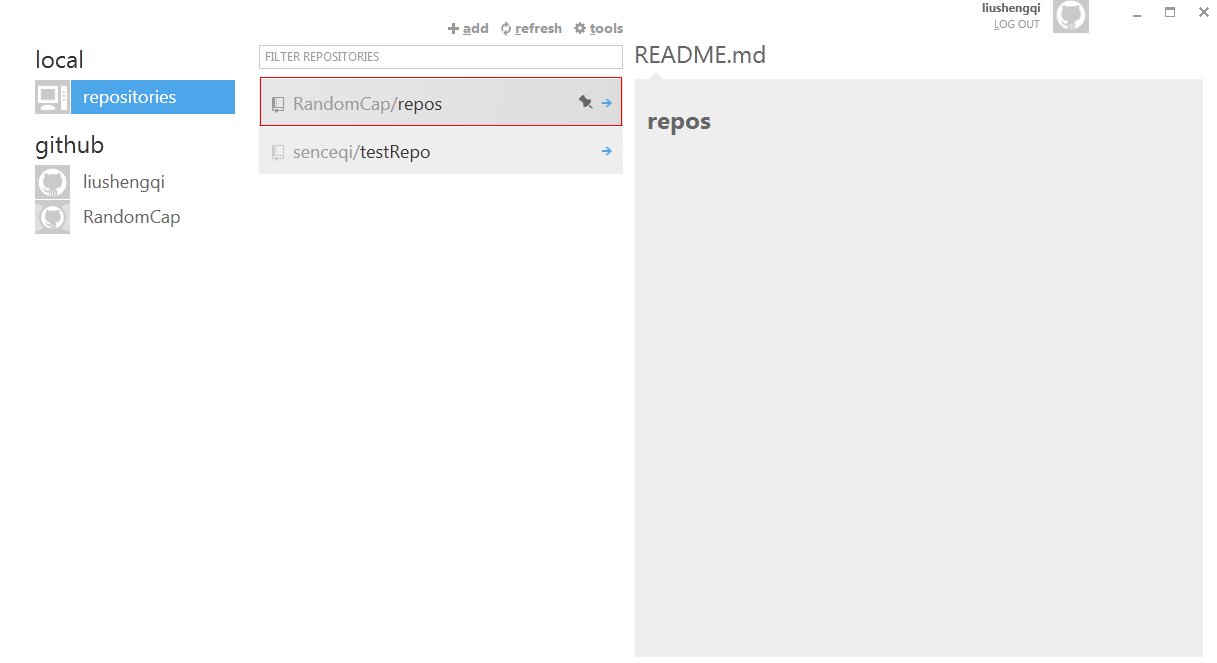
那么如何通过GitHub For Windows从GitHub clone下来数据仓库呢？

首先登录GitHub For Windows，然后在左侧找到需要clone的数据仓库，如图



**图3 Clone项目**

点击Clone图标，等待数据仓库被Clone下来，Clone完毕后点击local/repositories，将可以看到刚刚Clone到本地的数据仓库，到此Clone成功



**图4 Clone完成后**

### 3.2开发中的版本控制

Clone完成后，就可以将数据仓库中的项目文件导入到自己的开发环境中，开始开发工作了，开发工作中的版本控制主要涉及到增加文件、修改文件、删除文件。

#### 3.2.1版本控制流程

我们每天都需要提交我们所做到工作，以防止时间过长提交时造成更新过多，冲突难以解决。

使用Git作为版本控制系统，依据系统规模的大小，开发者工作职责的不同，开发者的版本维护工作流程也不相同，详细可见Git使用简介(第一版)。下面介绍我们一般开发中的版本控制模式，如下图



**图5 版本控制模式**

这种模式下，每个开发者都有Pull/Push权限。GitHub For Windows提交工作时工作流程，如图所示



**图5 GitHub For Windows 提交时工作流程**

相对应的Git For Windows 中使用Shell解决冲突的命令有：

|  |
| --- |
| //将gitHub上的数据仓库抓取到本地分支temp  git fetch git@github.com:repostofetch.git master:temp  //合并分支  git merge temp  //查看分支冲突的地方  git diff master temp  //解决冲突  手动找到冲突文件解决冲突  //add 所有文件  git add .  //提交  git commit |

到此使用Shell解决冲突完毕。可使用GitHub For Windows 同步到GitHub服务器。

* *Shell解决冲突中的 git add . git commit 也可以通过GitHub For Windows 来完成，推荐使用GitHub For Windows。同理也可以直接使用命令git push 来同步到远程服务器*

## 总结

GitHub是基于Git的，Git For Windows 也只是简化了操作，提升用户体验之举，最终底层还是使用Git命令和GitHub服务器交互，所以对Git的命令的了解对使用Git For Windows有更大的帮助。

以上只是根据个人使用经验寻求的一个使用GitHub For Windows来做版本控制的解决之道。相对于Git的强大功能来讲，只是冰山一角，有兴趣研究Git的可以深入研究，发现更好的方法及时交流。

## [参考文献]

1. GitHub官方介绍

https://help.github.com/articles/set-up-git

1. Git使用简介(第一版)