

title: Android基础入门教程——1.5.2 Git之使用GitHub搭建远程仓库

categories: Android

[基础入门, 教程]

Android基础入门教程——1.5.2 Git之使用GitHub搭建远程仓库

本节引言：

在上一节中，我们学习了如何使用Git，构建我们的本地仓库，轻松的实现了版本控制以及代码还原，修改日志查看等；读者肯定不满足与本地是吧，假如是多个人一起来开发一个程序呢？我们需要一个作为服务器的远程仓库！当然搭建一个服务器是需要成本的，为什么不把项目托管到Github上呢？作为开源代码库以及版本控制系统，Github拥有140多万开发者用户。随着越来越多的应用程序转移到了云上，Github已经成为了管理软件开发以及发现已有代码的首选方法，不需要任何成本，为何不使用呢？是吧！本节就来学习如何把我们的代码托管到Github上！

1.账号注册&仓库创建：

打开Github官网注册：[Github官网](#)，填写注册相关信息：用户昵称，邮箱，密码


注册完，跳转到如下页面，选择仓库购买方式(私有仓库,别人不可以访问，要权限)，一般我们自己玩选Free:

PS:对了，这时候你邮箱可能收到一封验证邮件，点下完成验证。


Plan	Cost (view in CNY)	Private repositories	
Large	\$50/month	50	<button>Choose</button>
Medium	\$22/month	20	<button>Choose</button>
Small	\$12/month	10	<button>Choose</button>
Micro	\$7/month	5	<button>Choose</button>
Free	\$0/month	0	 <button>Chosen</button>

接下来，创建一个我们的代码仓库：

Owner

 ZPJay /


Repository name

Garbage 


Great repository names are short and memorable. Need inspiration? How about **potential-octo-archer**.

Description (optional)

Test For Git

☒  **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: **Apache License 2.0** 


Create repository

为自己的仓库添加点内容提示，就是项目的一些概述(可写可不写)


 branch: master **Garbage** / + 

Initial commit

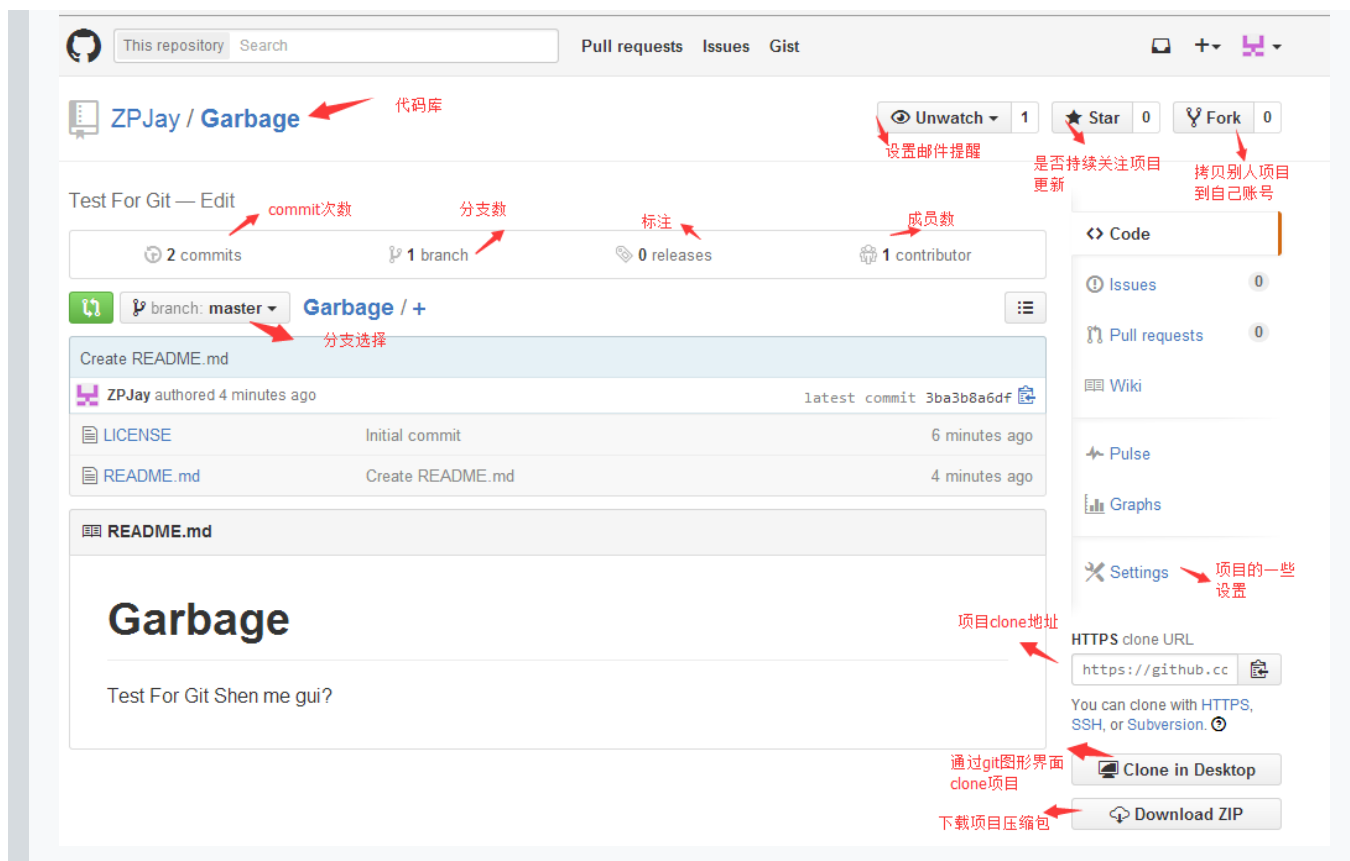
 ZPJay authored 2 minutes ago latest commit 372029c438 

 **LICENSE** Initial commit 2 minutes ago

Help people interested in this repository understand your project by adding a README!

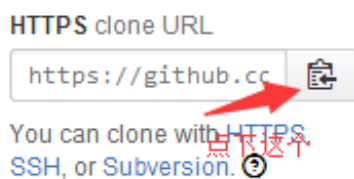


简单介绍下主页的一些东西：

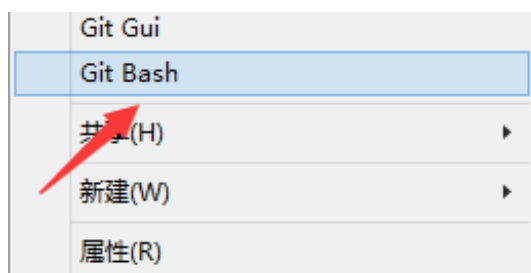


2.Clone代码库到本地

当然，你可以直接用图形化界面克隆，不过我还是喜欢通过命令行来Clone，先复制下Clone的地址



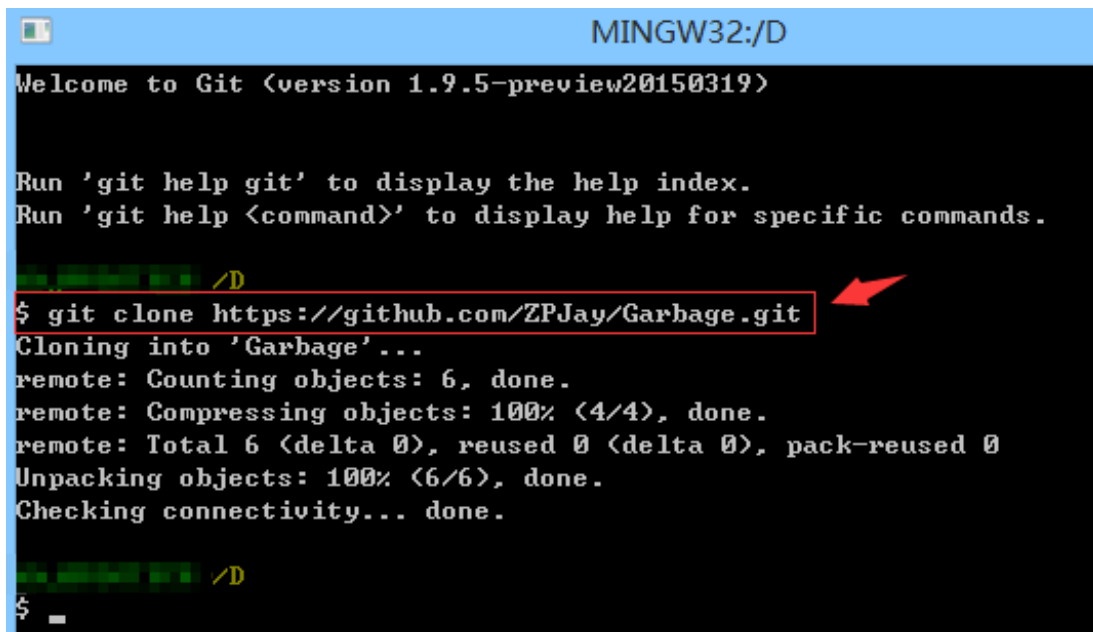
然后在某个地方，有键打开Git Bash：



键入：

```
git clone https://github.com/ZPJay/Garbage.git
```

然后可以看到我们的代码库就下载完成了：






```
MINGW32:/D
Welcome to Git (version 1.9.5-preview20150319)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

MINGW32 /D
$ git clone https://github.com/ZPJay/Garbage.git
Cloning into 'Garbage'...
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
Checking connectivity... done.

MINGW32 /D
$
```

打开文件夹，可以看到下述内容：

 .git	2015/6/24 星期...	文件夹	
 LICENSE	2015/6/24 星期...	文件	12 KB
 README.md	2015/6/24 星期...	MD 文件	1 KB



```
D:\Garbage\README.md - Notepad
文件(F) 编辑(E) 搜索(S) 视图(V) 格式(M) 语言(L) 设置(T) 宏(O) 运行(R) 插件(P) 窗口(W)
new 1x new 2x GetSchedule.java 中国城市代码.html README.md
1 # Garbage
2 Test For Git
3 Shen me gui?
```

3.分支管理

对于刚接触版本控制工具的朋友来说，分支可能比较陌生，但是他会给我们带来很大的便利！限于篇幅，笔者直接丢个链接，大家看看图就知道了：[曹雪峰的官方网站：创建和合并分支！](#)写得真心很赞~建议收藏！

了解概念后,我们来熟悉与分支相关的几个命令：

①创建分支(后者创建同时会切换分支):

```
git branch v1.0.3 或 git checkout -b v1.0.4
```

```
/D/Garbage <master>  
$ git checkout -b v1.0.4  
Switched to a new branch 'v1.0.4'
```

```
/D/Garbage <v1.0.4>  
$ git branch v1.0.3
```

②查看版本库中所有分支：

```
git branch -a
```

```
/D/Garbage <v1.0.4>  
$ git branch -a  
master  
v1.0.3  
* v1.0.4  
remotes/origin/HEAD -> origin/master  
remotes/origin/master
```

③切换到某一分支：

```
git checkout v1.0.3
```

```
/D/Garbage <v1.0.4>  
$ git checkout v1.0.3  
Switched to branch 'v1.0.3'  
  
/D/Garbage <v1.0.3>  
$ git checkout master  
Switched to branch 'master'  
Your branch is up-to-date with 'origin/master'.
```

④删除某一分支：

```
git branch -D v1.0.4
```

⑤合并分支

```
git merge v1.0.3
```

```
zpj@zpj: /D/Garbage (master)
$ git merge v1.0.3
Already up-to-date.
```

4.本地仓库与远程仓库同步问题

前面执行的这些分支操作都是在本地进行的，说了项目托管到GitHub上，我们肯定要跟远程仓库有交流是吧！

我们去年前面已经试过用clone命令把项目下载到本地，那么我们修改后如何把代码同步到Github上呢？我们先对我们的本地仓库做一点点修改，接着git add和git commit本地准备后，然后：

```
git push origin master 或者直接 git push
```

将我们本地的内容提交上去：

```
zpj@zpj: /D/Garbage (master)
$ git pull
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
From https://github.com/ZPJay/Garbage
   3ba3b8a..ed9e018  master       -> origin/master
   * [new branch]   v1.0.3       -> origin/v1.0.3
Already up-to-date!
Merge made by the 'recursive' strategy.
```

然后看下我们的Github,可以看到内容已经发生改变，而且提交者是我的另一个账号！

Test For Git — Edit

6 commits

2 branches

0 releases

1 contributor



branch: master ▾

Garbage / +

Merge branch 'master' of https://github.com/ZPJay/Garbage



coder-pig authored 8 minutes ago

latest commit 3c955dc

LICENSE

Initial commit

22 hou

README.md

Just for a test.

11 minut

README.md

Garbage

Test For Git Shen me gui? Silly B?

有同步到服务器，肯定有服务器同步到本地是吧，很简单，就一个

```
git pull
```

就可以了~

5.本节小结

好吧，本节就写那么多，相信你看到上面的Git教程还有一些冲突解决，分支管理，Bug分支等待高级的Git用法，考虑到这是入门教程，就不写那么深入了，有兴趣可以自己了解了解，说下自己公司目前的情况吧：

①使用Github作为我们的项目管理工具：我们都是把项目托管到Github上的，然后有两个分支：开发和测试两个分支，每个版本一个分支，最后发布时才把分支合并到master

上！提bug也是在上面提的，还是比较便利的！

②使用Trello来做流程控制，也是比较简洁高效的！有兴趣的可以了解了解！

另外，国内访问Github可能比较缓慢，而且如果是私有仓库是要收费的，如果公司没有使用代理或者是私人开发，可能略显鸡肋，不过可以考虑下使用国产的开源仓库：

Git@OSC，由开源中国提供的，提供了1000个私人仓库，好像，感觉还不错，有兴趣的可以考虑将代码托管到这里：<http://git.oschina.net/>！

就到这里，如果文中有错误纰漏，欢迎指出，谢谢~

——作者：**coder-pig**，本教程不收取任何费用，欢迎转载，尊重作者劳动成果，请勿用于商业用途，侵权必究！