**[Github教程](https://www.cnblogs.com/greyzeng/p/5046776.html)**

Git下载：<https://git-for-windows.github.io/>

我下载的版本是：Git-2.6.3-64-bit.exe

安装：略 默认选项点击"下一步"即可

安装完毕后

1.打开Git Bash

设置使用Git时候的名字和邮箱地址

$ git config --global user.name "yourname"

$ git config --global user.email "youremail@email.com"

2.注册Github账户：<https://github.com/>

3.设置SSH Key，在Git Bash中输入：

ssh-keygen –t rsa –C "github register email"

注: github register email这里写你在第二步注册Github账户的邮箱地址

然后按下回车，并设置认证密码（也可不设置）

回车，会得到两个文件：id\_rsa(私有密钥)，id\_rsa.pub是公开密钥。

这两个文件默认在C:\Users\Username\.ssh目录下

4.添加公开密钥：

进入你的Github账户，在右上角选择SettingSSH keysAdd SSH key, 其中，Title输入一个名称，在Key处粘贴id\_rsa.pub中的内容。

5.此时就可以用私人密钥和Github进行认证和通信，在Git Bash中输入：

ssh –T git@github.com

提示：Are you sure you want to continue connecting (yes/no)?

输入：yes 回车

显示：Hi yourname! You've successfully authenticated, but Github does not provide shell access.

接下来，演示一个Github的HelloWorld示例：

1. 进入Github账户，点击New repository
2. Repository name输入Hello
3. Description项输入一些对仓库的描述信息（选填）
4. Public/Private选项勾选Public
5. Initialize this repository with a README 选项选上
6. 点击Create Repository即可创建一个Repository
7. 点击进入Hello这个Repository，拷贝这个Repository的Web Address

https://images2015.cnblogs.com/blog/683206/201512/683206-20151215201918334-489121757.png

1. 将Hello这个Repository clone至本地，打开Git Bash，输入：

git clone your repository's Web Address

提示：repository's Web Address就是上一步骤拷贝的URL

1. Git Bash输入cd Hello，在Hello目录下增加一个文件，比如T.java
2. 将T.java添加到暂存区，Git Bash中输入：

git add T.java

1. 提交T.java, Git Bash中输入：

git commit –m "this is your comment"

1. Push到Github上的仓库

git push

进入Github账户中的Hello Repository，即可查看push进去T.java这个文件

1. 查看提交日志：

git log

Git基本命令

git diff：查看工作树，暂存区，最新提交之间的差别

举例:

在README.MD中增加一行

## This is the subtitle

执行git diff 命令，查看当前**工作树**和**暂存区**的区别。

[复制代码](javascript:void(0);)

$ git diff

diff --git a/README.md b/README.md

index fec5601..203cf31 100644

--- a/README.md

+++ b/README.md

@@ -1 +1,2 @@

# Hello

+## This is the subtitle

[复制代码](javascript:void(0);)

说明："+"表示新添加的行，"-"表示删除的行，由结果可知：增加了：## This is the subtitle

执行

git add README.md

将README.md文件加入暂存区

执行git diff HEAD 可以查看暂存区和最新提交的差别。

[复制代码](javascript:void(0);)

$ git diff HEAD

diff --git a/README.md b/README.md

index fec5601..203cf31 100644

--- a/README.md

+++ b/README.md

@@ -1 +1,2 @@

# Hello

+## This is the subtitle

[复制代码](javascript:void(0);)

执行

git commit –m "this is your description"

将文件提交。

git branch:显示分支列表，并标识当前所在分支

$ git branch

\* master

git checkout –b:创建并且切换分支

$ git checkout -b branch-a

M README.md

Switched to a new branch 'branch-a'

此时再运行git branch，显示当前分支为branch-a

$ git branch

\* branch-a

Master

在branch-a这个分支操作并提交代码，代码回提交到branch-a这个分支而不会提交到master分支。

举例：

在branch –a这个分支中修改README.md文件，增加一行：## this is branch-a 并提交，

[复制代码](javascript:void(0);)

$ git diff

diff --git a/README.md b/README.md

index fec5601..803aa74 100644

--- a/README.md

+++ b/README.md

@@ -1 +1,4 @@

# Hello

+## This is the subtitle

+

+## this is branch-a

[复制代码](javascript:void(0);)

切换至master分支

$ git checkout master

Switched to branch 'master'

Your branch is up-to-date with 'origin/master'.

查看master中的README.md文件，无增加行。

$ cat README.md

# Hello

git merge:合并分支

首先我们切换回master分支，

$ git checkout master

Switched to branch 'master'

Your branch is up-to-date with 'origin/master'.

然后，合并master和branch-a这个分支

执行：

git merge --no-ff branch-a

系统提示合并之前需要输入一些描述信息，按ESC并输入 :wq退出即可，

[复制代码](javascript:void(0);)

$ git merge --no-ff branch-a

Merge made by the 'recursive' strategy.

README.md | 3 +++

1 file changed, 3 insertions(+)

[复制代码](javascript:void(0);)

git log:查看历史提交记录

[复制代码](javascript:void(0);)

commit 182053ec7e8683e381996c683a7b04f43be57bd3

Merge: b1444e3 2806d72

Author: zenghui <zengh927@163.com>

Date: Tue Dec 15 19:31:02 2015 +0800

Merge branch 'branch-a'

commit 2806d720ed728ca55cc32c3091879eae8c8b5b5e

Author: zenghui <zengh927@163.com>

Date: Tue Dec 15 19:24:27 2015 +0800

add branch-a

commit b1444e36a2df109ce5414f5f00fd1ea97f3c1492

Author: zenghui <zengh927@163.com>

Date: Mon Dec 14 21:40:28 2015 +0800

Add T.java File

commit 0f2c6b68e711d74e35490bec6e6d8ab01e6a29d9

Author: GreyZeng <410486047@qq.com>

Date: Mon Dec 14 21:25:15 2015 +0800

[复制代码](javascript:void(0);)

git reset --hard + 哈希值：回溯到指定版本

执行：

$ git reset --hard b1444e36a2df109ce5414f5f00fd1ea97f3c1492

HEAD is now at b1444e3 Add T.java File

回溯到branch-a分支创建之前。

处理冲突

举例：

在master的README.md中增加一行：## FROM MASTER并提交

合并master和branch-a分支

[复制代码](javascript:void(0);)

$ git merge --no-ff branch-a

Auto-merging README.md

CONFLICT (content): Merge conflict in README.md

Automatic merge failed; fix conflicts and then commit the result.

[复制代码](javascript:void(0);)

显示冲突

用编辑器打开README.md文件发现:

# Hello

<<<<<<< HEAD

## FROM MASTER

=======

## This is the subtitle

## this is branch-a

>>>>>>> branch-a

此时，只要手动修改一下文件：

# Hello

## FROM MASTER

## This is the subtitle

## this is branch-a

再次执行git add 和 git commit 即可

git push:推送至远程仓库

推送master到远程仓库：

[复制代码](javascript:void(0);)

$ git push -u origin master

Counting objects: 9, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (8/8), done.

Writing objects: 100% (9/9), 889 bytes | 0 bytes/s, done.

Total 9 (delta 0), reused 0 (delta 0)

To https://github.com/GreyZeng/Hello.git

b1444e3..6840c0a master -> master

Branch master set up to track remote branch master from origin.

[复制代码](javascript:void(0);)

也可以推送某个分支到远程仓库

比如：推送branch-a到远程仓库,需要执行以下命令：

[复制代码](javascript:void(0);)

$ git checkout branch-a

Switched to branch 'branch-a'

$ git push -u origin branch-a

Total 0 (delta 0), reused 0 (delta 0)

To https://github.com/GreyZeng/Hello.git

\* [new branch] branch-a -> branch-a

Branch branch-a set up to track remote branch branch-a from origin.

[复制代码](javascript:void(0);)

git clone:获取远程仓库

首先我们切换到其他目录下：

执行git clone <https://github.com/GreyZeng/Hello.git>

[复制代码](javascript:void(0);)

$ git clone https://github.com/GreyZeng/Hello.git

Cloning into 'Hello'...

remote: Counting objects: 15, done.

remote: Compressing objects: 100% (12/12), done.

remote: Total 15 (delta 0), reused 12 (delta 0), pack-reused 0

Unpacking objects: 100% (15/15), done.

Checking connectivity... done.

[复制代码](javascript:void(0);)

从而获取到远程仓库

同时，我们可以从远程仓库获取之前推送的branch-a这个分支

[复制代码](javascript:void(0);)

$ git checkout -b branch-a origin/branch-a

Branch branch-a set up to track remote branch branch-a from origin.

Switched to a new branch 'branch-a'

[复制代码](javascript:void(0);)

git pull:获取最新的远程仓库：

假如有人在master分支上push了新的代码，本地需要同步最新代码

[复制代码](javascript:void(0);)

$ git pull origin master

remote: Counting objects: 3, done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

From https://github.com/GreyZeng/Hello

\* branch master -> FETCH\_HEAD

6b04921..c2c9aa0 master -> origin/master

Updating 6b04921..c2c9aa0

Fast-forward

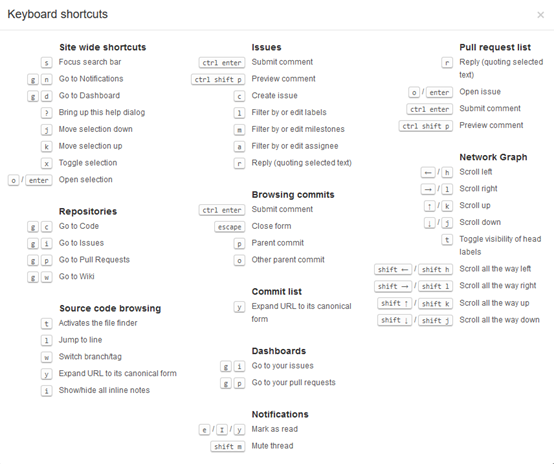
README.md | 2 +-

1 file changed, 1 insertion(+), 1 deletion(-)

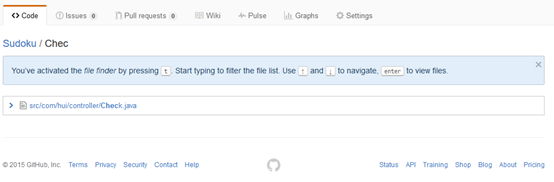
一些小技巧：

在Github的个人主页中按Shift+/

可以显示快捷键的操作：

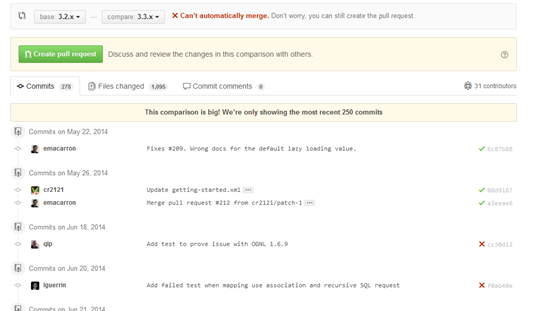


在某个项目的文件列表中，按下t，即可根据输入的文件名和部分文件名查找文件：



对比两个分支之间的差别，以[mybatis](https://github.com/mybatis/mybatis-3)项目为例，要对比3.2.x和3.3.x两个分支之间的差别，如下格式：

<https://github.com/mybatis/mybatis-3/compare/3.2.x...3.3.x>



类似地:

查看master分支最近7天内的差别：

[https://github.com/mybatis/mybatis-3/compare/master@{7.day.ago}...master](https://github.com/mybatis/mybatis-3/compare/master@%7b7.day.ago%7d...master)

查看与指定日期之间的差别：

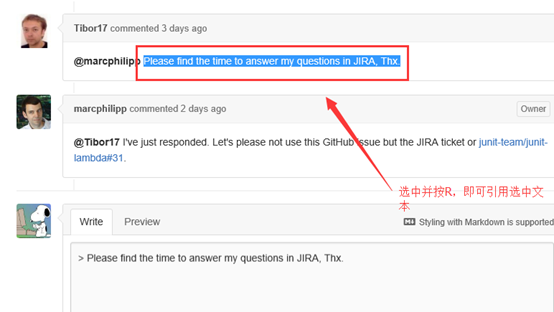
[https://github.com/mybatis/mybatis-3/compare/master@{2015-09-27}...master](https://github.com/mybatis/mybatis-3/compare/master@%7B2015-09-27%7D...master)

用户可以提交Issue，并把Issue标记为bug，在提交代码的时候，加上以下描述信息，对应的Issue就会被Close

fix #+Issue编号

如：fix #1 对应编号为1的Issue被关闭。

只要选中评论的文本，并按下R，即可引用选中文本



Pull Request

Pull Request 是自己修改源代码后，请求对方仓库采纳该修改时采取的一种行为。

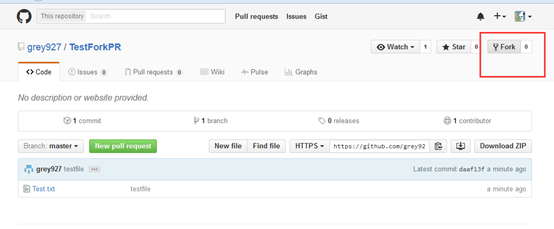
场景1：

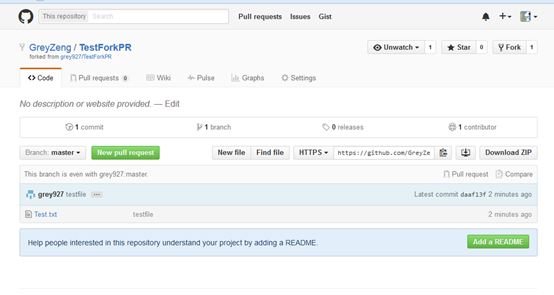
用户A在fork完用户B的项目时，A修改了代码并提交了一个Pull Request给B用户，B用户Merge了A用户提交的Pull Request。

假设需要fork以下项目：

<https://github.com/grey927/TestForkPR>

点击账户右上角的fork按钮，fork项目到自己账户下





打开Git Bash

将项目clone到本地：

[复制代码](javascript:void(0);)

$ git clone https://github.com/GreyZeng/TestForkPR.git

Cloning into 'TestForkPR'...

remote: Counting objects: 3, done.

remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

Checking connectivity... done.

[复制代码](javascript:void(0);)

进入TestForkPR目录，查看分支状态：

[复制代码](javascript:void(0);)

$ git branch -a

\* master

remotes/origin/HEAD -> origin/master

remotes/origin/master

[复制代码](javascript:void(0);)

在TestForkPR目录下增加一个文件，T.java

增加到暂存区：

$ git add T.java

提交修改：

[复制代码](javascript:void(0);)

$ git commit -m "this is test"

[master 2b72d0f] this is test

1 file changed, 5 insertions(+)

create mode 100644 T.java

[复制代码](javascript:void(0);)

Push到自己账户下的远程仓库下：

[复制代码](javascript:void(0);)

$ git push -u origin master

Counting objects: 3, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 345 bytes | 0 bytes/s, done.

Total 3 (delta 0), reused 0 (delta 0)

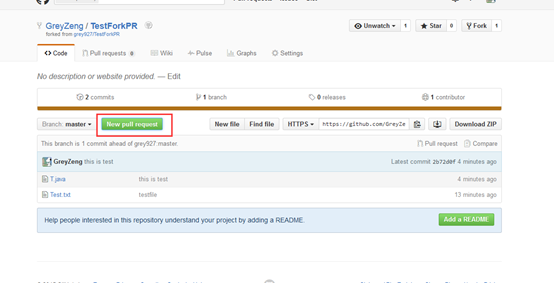
To https://github.com/GreyZeng/TestForkPR.git

daaf13f..2b72d0f master -> master

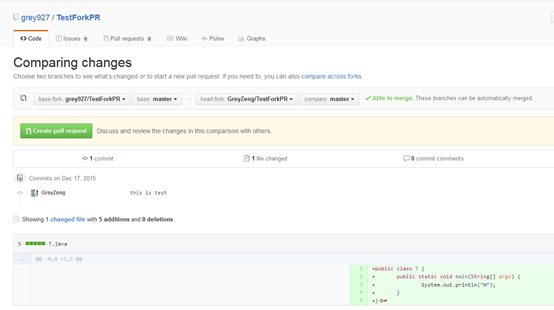
Branch master set up to track remote branch master from origin.

[复制代码](javascript:void(0);)

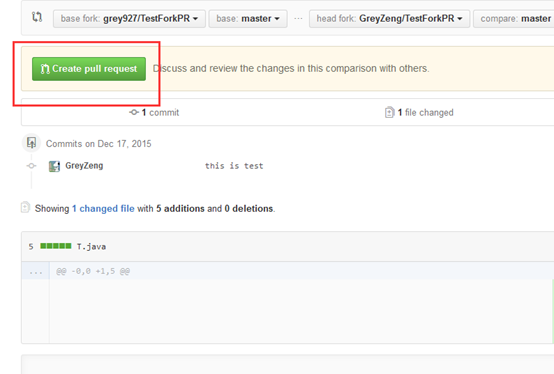
打开自己账户的远程仓库，点击New Pull Request按钮：



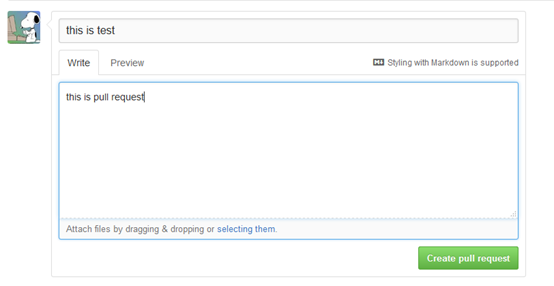
弹出文件差异对比界面



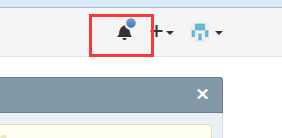
确认无误时，点击Create pull request

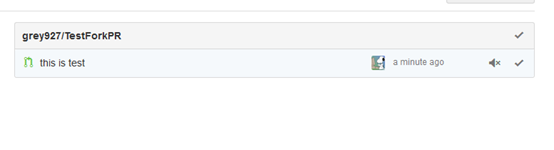


写上相关描述并点击Create pull request:

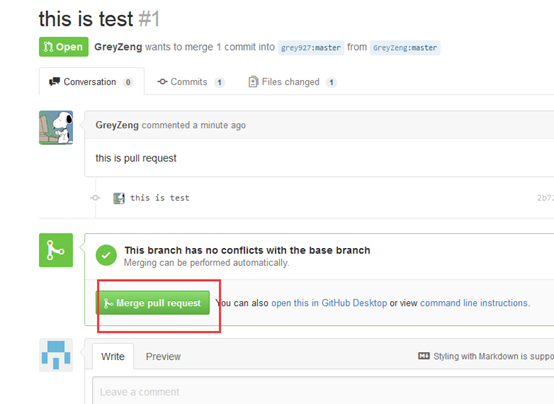


此时，原仓库会收到一条pull request的信息：

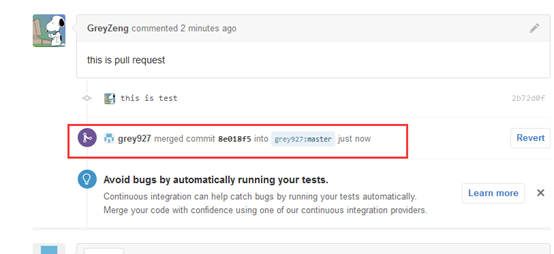




点击进入这条信息，



点击Merge pull request，即可接受这个pull request的请求。



场景2：

直接clone原仓库到本地，不执行fork操作，并同时使用fetch/merge命令使本地仓库和原仓库保持代码一致。

举例：

原仓库的地址：<https://github.com/grey927/TestFetchMerge>

直接clone至本地：

[复制代码](javascript:void(0);)

$ git clone https://github.com/grey927/TestFetchMerge.git

Cloning into 'TestFetchMerge'...

remote: Counting objects: 3, done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

Checking connectivity... done.

[复制代码](javascript:void(0);)

给原仓库设置upstream的名称，将其作为远程仓库。

$ git remote add upstream https://github.com/grey927/TestFetchMerge.git

我们每次只要从原仓库获取最新源码，并和本地分支进行合并即可：

获取最新源码：

[复制代码](javascript:void(0);)

$ git fetch upstream

remote: Counting objects: 3, done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

From https://github.com/grey927/TestFetchMerge

\* [new branch] master -> upstream/master

[复制代码](javascript:void(0);)

合并：

[复制代码](javascript:void(0);)

$ git merge upstream/master

Updating e2215cc..a215a8a

Fast-forward

newfile.txt | 1 +

1 file changed, 1 insertion(+)

create mode 100644 newfile.txt