Git分支管理

将git的分支分为3类，主分支(master)、工作分支(develop)、临时分支。

主分支和工作分支一直存在。

临时分支用完后删除。

吾问无为谓无无无无无

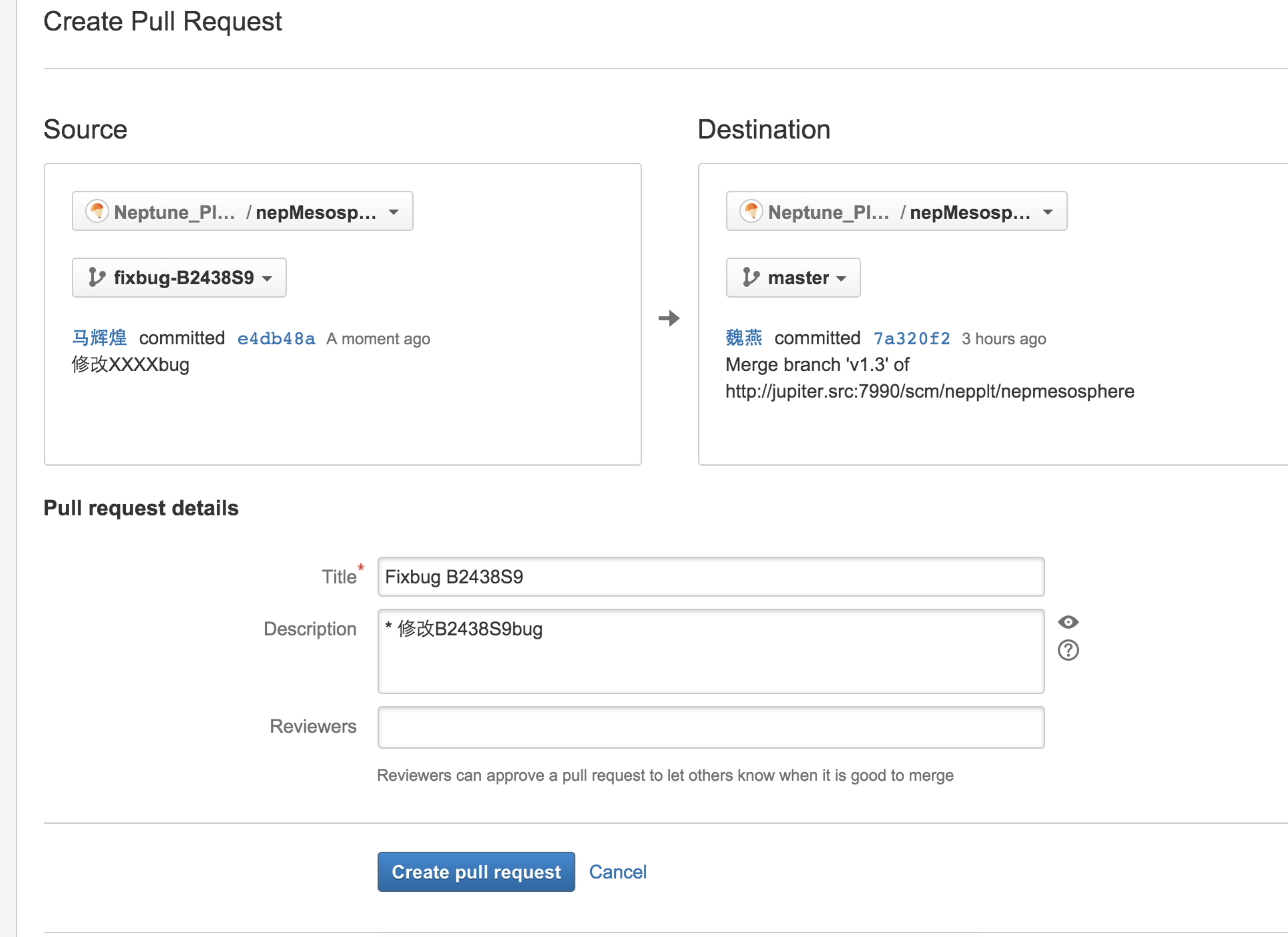
# 主分支

主分支：master。

主分支提供给所用用户正式版，都在主分支上发布。

master上获取的代码是随时都可以运行的。

目前，使用Pull request将临时分支合并到主分支。



# 工作分支

工作分支：develop

日常的开发在develop上完成。

# 临时分支

临时分支用于应对一些特定目的的开发。

一个临时分支只对应一个功能开发或者一个bug的修复。

目前，我们使用三种临时分支：功能分支、预发分支、修复分支。

* **功能分支：**

功能分支命名方式：feature-\*\*\*

功能分支是为了开发摸个特定的功能，从develop上面分出来，开发完成后，要并入到develop。

|  |
| --- |
| 创建一个功能分支：  **git checkout –b** *feature-adduserinfo* **develop**  开发完成能后， 将功能分支合并到develop分支：  **git checkout develop**  **git merge –no-ff** *feature-adduserinfo*  再删除功能分支  **git branch -d**  *feature-adduserinfo* |

* **预发分支**

预发分支命名方式：*release* -\*\*\*

预发布分支，它是指发布正式版本之前（即develop合并到master分支之前），我们可能需要有一个预发布的版本进行测试。

预发布分支是从develop分支上面分出来的，预发布结束以后，必须合并进develop和master分支。

|  |
| --- |
| 创建一个预发分支：  **git checkout –b** *release-1.3.3* ***develop***  测试完成并修改bug确认没有问题后， 将预发分支合并到master分支：  **git checkout *master***  **git merge –no-ff** *release-1.3.3*  再合并到develop分支  **git checkout *develop***  **git merge –no-ff** *release-1.3.3* |

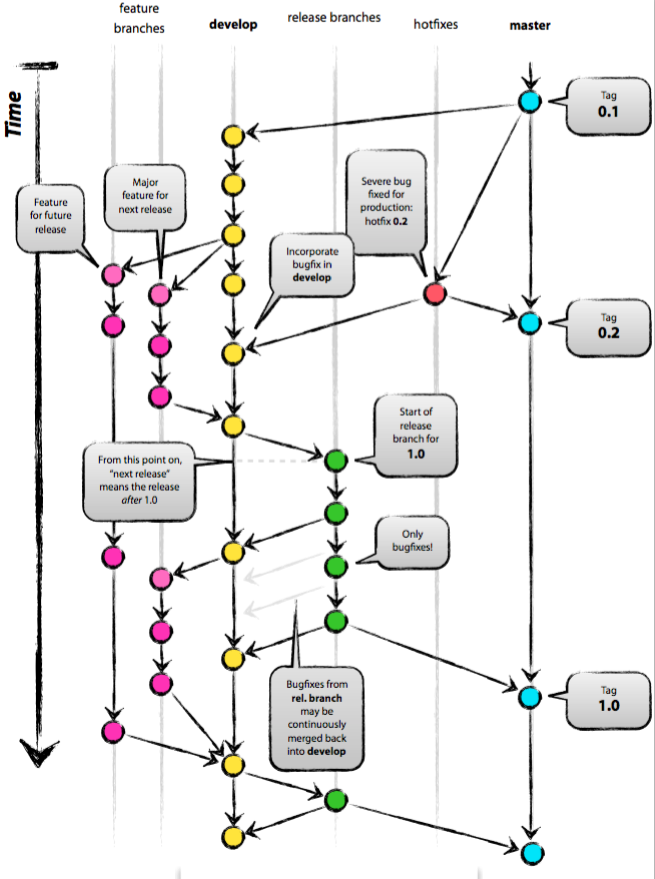
* **修补分支**

修补分支的命名方式：fixbug-\*\*\*

修补分支就是用来修改bug。

修补bug分支是从Master分支上面分出来的。修补结束以后，再合并进Master和Develop分支。

|  |
| --- |
| 创建一个修补分支：  **git checkout -b** *fixbug-0.1* ***master***  修补结束后，合并到master分支：  **git checkout** ***master***  **git merge --no-ff** *fixbug-0.1*  再合并到develop分支  **git checkout *develop***  **git merge --no-ff**  *fixbug-0.1*  最后删除修补分支  **git branch** *fixbug-0.1* |



# [Git 获取远程分支](http://www.cnblogs.com/hanxianlong/archive/2012/09/10/2678659.html)

另一哥们将分支push到库中，我怎么获取到他的分支信息呢？

如果安装了git客户端，直接选择fetch一下，就可以获取到了。

如果用命令行，运行 git fetch，可以将远程分支信息获取到本地，再运行 git checkout -b local-branchname origin/remote\_branchname  就可以将远程分支映射到本地命名为local-branchname  的一分支。

[**git 删除远程分支**](http://www.cnblogs.com/shiningrise/archive/2013/03/12/2956779.html)

一不小心把本地的临时分支push到server上去了，想要删除。  
一开始用  
  
git branch -r -d origin/branch-name  
  
不成功，发现只是删除的本地对该远程分支的track，正确的方法应该是这样：  
  
git push origin :branch-name  
  
冒号前面的空格不能少，原理是把一个空分支push到server上，相当于删除该分支。