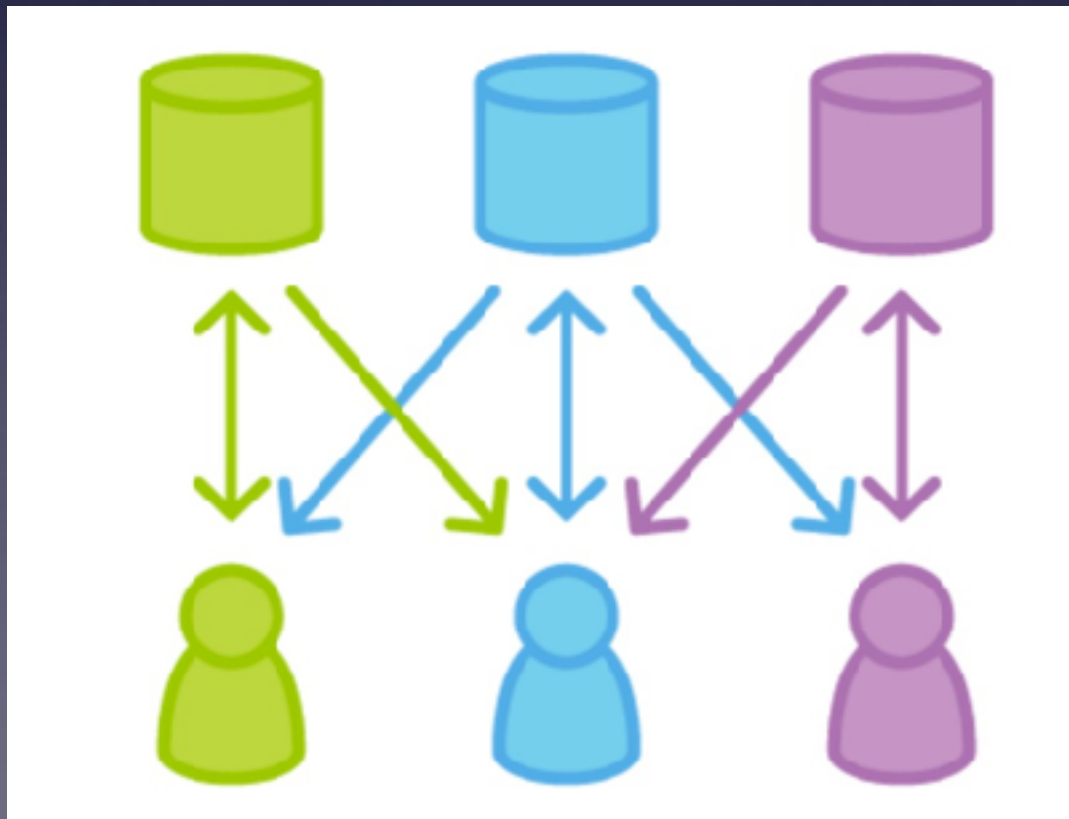


# Git Forking工作流的分享

作者：应用产品中心-王燦

# 什么是Forking流

Forking这种工作流不是使用单个服务端仓库作为『中央』代码基线，而让各个开发者都有一个服务端仓库。这意味着各个代码贡献者有2个Git仓库而不是1个：一个本地私有的，另一个服务端公开的。

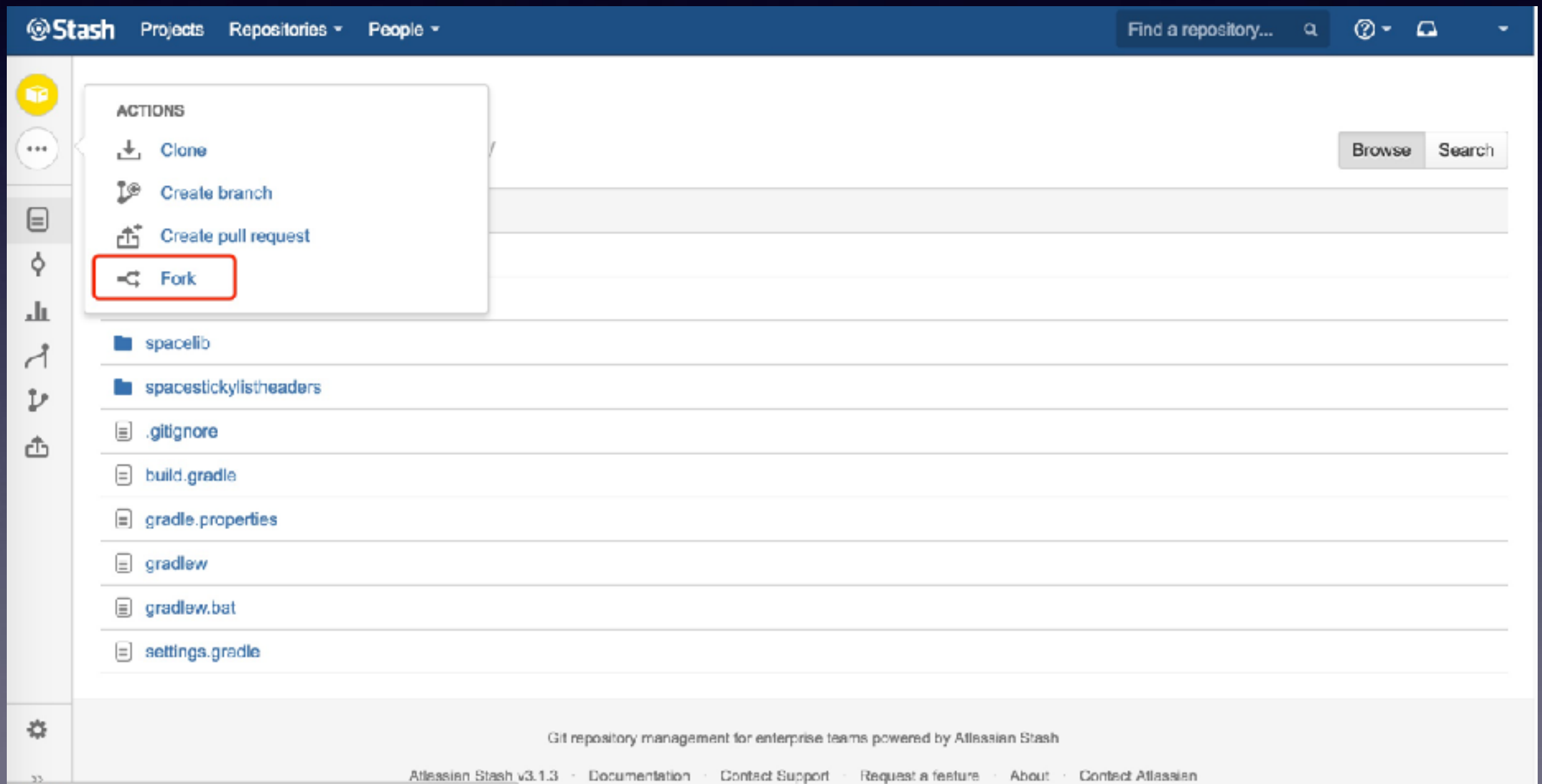


# 工作方式

- 1.和其它的Git工作流一样，Forking工作流要先有一个公开的正式仓库存储在服务器上。但一个新的开发者想要在项目上工作时，不是直接从正式仓库克隆，而是fork正式项目在服务器上创建一个拷贝。
- 2.这个仓库拷贝作为他个人公开仓库——其它开发者不允许push到这个仓库，但可以pull到修改（后面我们很快就会看这点很重要）。在创建了自己服务端拷贝之后，和之前的工作流一样，开发者执行git clone命令克隆仓库到本地机器上，作为私有的开发环境。
- 3.要提交本地修改时，push提交到自己公开仓库中——而不是正式仓库中。然后，给正式仓库发起一个pull request，让项目维护者知道有更新已经准备好可以集成了。对于贡献的代码，pull request也可以很方便地作为一个讨论的地方。
- 4.为了集成功能到正式代码库，维护者pull贡献者的变更到自己的本地仓库中，检查变更以确保不会让项目出错，合并变更到自己本地的master分支，然后pushmaster分支到服务器的正式仓库中。到此，贡献的提交成为了项目的一部分，其它的开发者应该执行pull操作与正式仓库同步自己本地仓库。

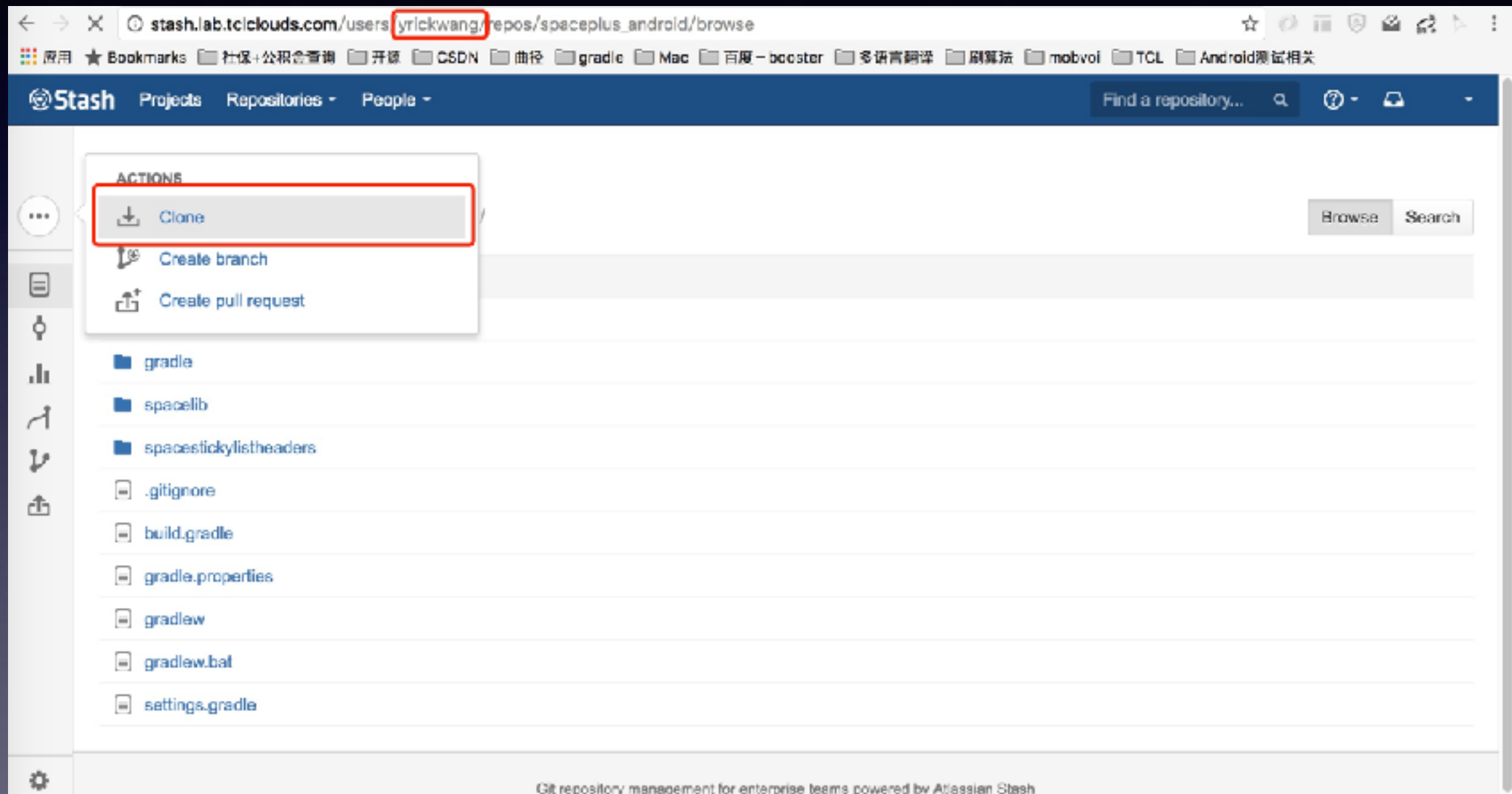
# 具体步骤1

开发者fork正式仓库



# 具体步骤2

开发者克隆自己fork出来的仓库到本地(如果已经clone过正式仓库, 就直接进行2.1的操作)



2.1在项目里面添加新的remote (PS: upstream是自己定义起的名字, 可以随意的 ssh地址也是根据自身情况替换的)

有一点需要注意下: 如果点击clone找不到ssh的话说明你没有在stash上面添加token, 在自己的MyActivity设置模块去添加  
git remote add upstream ssh://git@stash.lab.tclclouds.com:7999/~yrickwang/spaceplus\_android.git

# 具体步骤3

开发者在本地开发自己的功能

这个时候就按之前的操作一样，正常操作，比如编辑代码、提交修改、新建分支

# 具体步骤4

开发者push自己的commit（请严格按照如下3步进行操作）

- 1.git remote update(进行仓库同步，即将正式仓库的代码同步到我们的fork仓库里，注意：这里本地的仓库不会发生任何变化的)
- 2.git merge/rebase upstream master(进行代码合并，这里其实就是把fork仓库里的代码同步到本地)
  - 特别的需要注意下：
    - \*当本地没有任何commit的时候，采用merge进行合并代码
    - \*当本地有commit的时候，进行rebase合并代码
- 3.git push upstream master(代码提交到fork仓库)

# 具体步骤5

## 开发者发起pull request

The screenshot shows the Stash web interface for a repository named 'spaceplus\_android'. The 'Pull requests' section is active, displaying a guide titled 'Getting started with pull requests'. The guide consists of three steps: 'Branch', 'Discuss', and 'Merge'. A red arrow points from the 'Create a pull request' button to the 'Branch' step. The 'Branch' step includes a diagram of a branch and the text: 'Develop features on a branch and create a pull request to get changes reviewed.' The 'Discuss' step includes a diagram of a pull request with comments and the text: 'Discuss and approve code changes related to the pull request.' The 'Merge' step includes a diagram of a merge and the text: 'Merge the branch with the click of a button. [Learn about pull requests.](#)'

Stash Pull requests

Open Merged Declined

Getting started with pull requests [Create a pull request](#)

**Branch**  
Develop features on a branch and create a pull request to get changes reviewed.

**Discuss**  
Discuss and approve code changes related to the pull request.

**Merge**  
Merge the branch with the click of a button.  
[Learn about pull requests.](#)

正在连接...




# 具体步骤6

开发者create request 并且添加代码审核者reviewer

### Create Pull Request

#### Source

选择你的fork仓库的分支

 yrickwan / spaceplus\_... ▾


master ▾

**chaohao.zhou** committed [71db80f](#) 1 min ago

修改数据库文件，保证暂时可以扫出数据

#### Destination

选择你要合入的正式仓库分支

 SOG-Space / spaceplus\_... ▾

master ▾

**chaohao.zhou** committed [71db80f](#) 1 min ago

修改数据库文件，保证暂时可以扫出数据


#### Pull request details

Title \*

Master

Description

添加你的代码审核者



Reviewers

chaohao.zhou x

# 具体步骤7

有人提交了Pull Request之后，reviewer会收到邮件通知，我们进入如下界面查看commit

**SOG-Space Clean**  
spaceplus\_android

**ACTIONS**

- Clone
- Create branch
- Create pull request
- Fork

**NAVIGATION**

- Source
- Commits
- Graphs
- Network
- Branches
- Pull requests** 1

**Pull requests**

查看已经merge过的commit

Open Merged Declined 查看已经拒绝过的commit

ID	Title	Author	Reviewers	Source	Destination	Updated
#31	Master	yrickwang		spaceplus... b m...	b master	Yesterday

查看该commit的具体

若有新的pull request，这里就会有提示+1

# 具体步骤8

审核具体的commit，详细的介绍直接看下图

The screenshot shows the Stash web interface for a pull request. The browser tabs at the top include 'mweb for', 'MWeb', '知乎 mac', 'Mac 上通', 'Releases', 'git fork', 'github fork', 'Git 工作流', '百度一下', 'Pull Request', and 'wong'. The address bar shows the URL: 'stash.lab.telclouds.com/projects/CLEAN/repos/spaceplus\_android/pull-requests/31/diff'. The Stash logo and navigation links (Projects, Repositories, People) are at the top. The left sidebar shows the repository 'SOG-Space Clean spaceplus\_android' and various actions like Clone, Create branch, Create pull request, and Fork. The main area is titled 'Pull request #31 OPEN' and shows a diff between 'spaceplus...' and 'master'. The 'Commits' tab is selected, and the file 'SpaceApplication.java' is highlighted. The diff view shows changes to 'SpaceApplication.java', with line numbers 49 to 69. A red box highlights the code diff, and a red arrow points to the comment area. The 'Merge' and 'Decline' buttons are visible at the top right.

Stash Projects Repositories People Find a repository...

SOG-Space Clean spaceplus\_android

ACTIONS

- Clone
- Create branch
- Create pull request
- Fork

NAVIGATION

- Source
- Commits
- Graphs
- Network
- Branches
- Pull requests 1

Settings

Pull request #31 OPEN spaceplus... master

Master

Overview Diff Commits

Find text in diff and context lines

app/src/main/java/com/clean/spaceplus/app

SpaceApplication.java

在这个区域可以逐行的进行代码评论，例如指出哪里不合理，或者提出更好的解决方案

app/src/main/java/com/clean/spaceplus/app/SpaceApplication.java MODIFIED

```
49 49 private static SpaceApplication instance;
50 50 public static SpaceApplication getInstance(){
51 51     return instance;
52 52 }
53 53 private static Tracker mTracker;
54 54 private Handler mHandler = null;
55 55 @Override
56 56 public void onCreate() {
57 57     super.onCreate();
58 58     // LeakCanary.install(this); //内存泄露检测
59 59     //asdasdasds
60 59 instance = this;
61 60
62 61 //启动服务
63 62 SecularService.launch(-1);
64 63
65 64 if(BuildConfig.DEBUG){
66 65     setDeveloperMode();
67 66 }
68 67 init(this);
69 68 // 配置异常处理类
```

觉得代码没有问题就合并到正式仓库 Merge Decline Edit

觉得不合理，就拒绝本次commit

# 一些特殊情景

## A.origin仓库与upstream仓库不同步的情况 (此情景非常经典)

举例：本来origin和upstream是同步的，随着开发周期迭代，本地(native)里面有很多commit已经提交到upstream了，然后呢很久没有pr，当然，origin也是会有很多新的commit，然后突然某个时刻，开发同学想起来有很多commit忘记pr了，这个时候准备先更新再pr，结果发现更新不过来，说有冲突（PS:这里的冲突描述下，原因是因为你的upstream和origin是完全脱节了，即origin和upstream里面都有各自没有的新的commit），然后也没有办法pr,形成了一个死循环，这种情况怎么处理？

具体操作如下：

- 1.先从origin上面同步代码到本地(native)，`git rebase origin/$branch` 然后肯定有冲突，我们就解决冲突。
- 2.发push到upstream 这个时候是肯定需要加-f的 (强推)
- 3.最后发pr到origin 就行了

以上就是本人对forking流的一些见解，有什么说不到位或者没说到的地方，请大家直接去多联系多操作，多使用-help命令。欢迎大家加我交流。

谢谢！

email: [yrickwang@tcl.com](mailto:yrickwang@tcl.com)