

如何给Apache Flink贡献你的代码

Apache

Flink开源大数据处理系统最近比较火，特别是其流处理框架的设计。本文并不打算介绍Apache Flink的相关概念，如果你感兴趣可以到本博客的[Flink分类目录](#)查看Flink的相关文章。

转入正题了，下面将一步一步教你如何提交你的代码到Flink社区。

1、提交Issue

既然能够

提交代码肯定是发

现了什么Bug，或者有什么好的新功能

，所以我们需要先到

<https://issues.apache.org/jira/browse/FLINK>

页面提交相关的Issue，说明这个Bug出现的现象，或者新功能的相关信息，比如我之提交Issue：[FLINK-5058](#)

2、Fork Apache Flink代码

到 [Flink项目的GitHub页面](#) 的右上角点击 Fork按钮

；然后你

将可以到你的GitHub

页面看到Fork好的代码，比如我Fork之后的页面地址是：<https://github.com/397090770/flink>

3、同步代码到本地

在本地使用git将Fork好的代码同步到你本地的某个文件夹下：

```
$ git clone https://github.com/397090770/flink.git
```

这时候会在当前目录下产生一个名为flink的文件夹

4、添加远程库地址

使用下面地址可以添加Flink的远程库地址

```
git remote add upstream https://github.com/apache/flink.git
```

因为我们Fork代码之后，Flink的主分支的代码可能已经发生了变化，为了避免冲突，我们需要同步一下Flink代码：

```
git pull upstream master
```

现在你本地master分支的代码和Apache上Flink的master分支代码一致了。

5、创建分支

我们一般会为一个新增的功能或者bug修复新增一个新的分支(branch)，可以使用下面命令创建：

```
git checkout -b flink-5058
```

6、添加自己的代码

我们现在可以使用自己喜欢的开发工具（比如Idea、Eclipse等）添加代码，之后我们使用下面的命令提交新增的代码了，不过在我们添加代码的时候，Apache上的Flink代码可能已经发生了变更，我们需要再一次同步，因为我们上面已经在 flink-5058 分支上了，我们需要先切换到我们自己Fork的代码master分支上，然后同步Apache上的Flink主分支代码到本地master分支：

```
git checkout master  
git pull upstream master
```

同步完之后，可以提交刚刚我们自己添加的代码了，如下：

```
git checkout flink-5058  
git add flink-scala-shell  
git commit -m "taskManagerMemory attribute set wrong value"  
git rebase master  
git push origin flink-5058
```

第一个是切换到 flink-5058 分支；第二个是添加我们刚刚修改代码的目录；第三个是添加注释，

这个一定要写；第四个是分支的行合（具体啥含义看[这里](#)）；第五个提交代码到flink-5058分支。

7、提交PR

PR的全称是Pull Request。我们提交完代码到flink-5058分支分支之后，就可以到<https://github.com/397090770/flink/pulls>页面的按下 New pull request 按钮；然后按照要求填写好标题，内容，最后点击 Create pull request 按钮就可以提交你刚刚修改的代码到Flink社区。我提交的RP地址：<https://github.com/apache/flink/pull/2799>（写的不太好）

注意：填写PR标题的格式是 [JIRA序号]+标题，比如我的PR标题是 [FLINK-5058]taskManagerMemory attribute set wrong value。FLINK-5058就是第一步创建的Issue序号。这样系统可以自动将RP关联到Issue上面。

8、等待审核

提交完PR之后，需要等待社区审核；必要时需要在提交的PR页面讨论。如果顺利的话，社区会将你的代码合并到Apache的Flink主分支。然后之前提交的Issue状态将会变成CLOSED，并且会显示在Flink的什么版本解决。

好了，到这里你已经学会了贡献代码的所有步骤。

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接: [【】（）](#)