

# Git 实践

Bruce

January 20, 2018

## Contents

<b>1</b>	<b>Git 配置</b>	<b>2</b>
1.1	设置用户名和邮件地址 . . . . .	2
1.2	设置别名 . . . . .	2
1.3	开启颜色显示 . . . . .	2
1.4	删除配置 . . . . .	2
<b>2</b>	<b>常用命令</b>	<b>2</b>
2.1	暂存区 . . . . .	2
2.2	重置 . . . . .	2
2.3	检出 . . . . .	2
<b>3</b>	<b>最佳实践</b>	<b>2</b>
3.1	备份 git 库 . . . . .	2
3.2	处理分离头指针 . . . . .	2

## 1 Git 配置

### 1.1 设置用户名和邮件地址

### 1.2 设置别名

### 1.3 开启颜色显示

### 1.4 删除配置

## 2 常用命令

### 2.1 暂存区

### 2.2 重置

### 2.3 检出

## 3 最佳实践

### 3.1 备份 git 库

场景: 当已经存在一个 git 库, 为了防止代码库被误删, 比较好的做法就是在本地建立一个备份, 并经常与代码库同步。步骤如下: 假设已经存在 git 代码库为 [/home/caogh1/workspace/hub], 备份库位置:[/media/back]

<1> 使用 git init 命令创建一个空的裸版本库

```
git init --bare /media/back/hub.git
```

<2> 使用 push 命令为其创建内容

```
cd /home/caogh1/workspace/hub
git push --set-upstream /media/back/hub.git master
```

<3> 更新工作区内容, 使用 git push 与备份库同步

```
git push
```

### 3.2 处理分离头指针

场景: 当 HEAD 处于分离头指针时, 处理如下。”分离头指针”状态指的就是 HEAD 头指针指向了一个具体的提交 ID, 而不是一个引用 (分支)。

`git checkout master`

如果在”分离头指针模式”下有提交，不能通过 `master` 分支或其他引用访问。如果提交是 `master` 分支所需要的，可以使用合并操作 (`git merge`)，具体操作过程如下：

<1> 确认当前处于 `master` 分支；

```
git branch -v
```

<2> 执行合并操作，将提交合并到当前分支；

```
git merge <id>
```

<3> 仔细看看最新提交，会看到这个提交有两个父提交，这就是合并的奥秘。

```
git cat-file -p HEAD
```