Git 查看提交历史 Git 远程仓库(Github)

Git 标签

如果你达到一个重要的阶段，并希望永远记住那个特别的提交快照，你可以使用 git tag 给它打上标签。

比如说，我们想为我们的 runoob 项目发布一个"1.0"版本。 我们可以用 git tag -a v1.0 命令给最新一次提交打上（HEAD）"v1.0"的标签。

-a 选项意为"创建一个带注解的标签"。 不用 -a 选项也可以执行的，但它不会记录这标签是啥时候打的，谁打的，也不会让你添加个标签的注解。 我推荐一直创建带注解的标签。

```
$ git tag -a v1.0
```

当你执行 git tag -a 命令时，Git 会打开你的编辑器，让你写一句标签注解，就像你给提交写注解一样。

现在，注意当我们执行 git log --decorate 时，我们可以看到我们的标签了：

```
$ git log --oneline --decorate --graph
*   88afe0e (HEAD, tag: v1.0, master) Merge branch 'change_site'
|\
| * d7e7346 (change_site) changed the site
* | 14b4dca 新增加一行
|/
* 556f0a0 removed test2.txt
* 2e082b7 add test2.txt
* 048598f add test.txt
* 85fc7e7 test comment from runoob.com
```

如果我们忘了给某个提交打标签，又将它发布了，我们可以给它追加标签。

例如，假设我们发布了提交 85fc7e7(上面实例最后一行)，但是那时候忘了给它打标签。 我们现在也可以：

```
$ git tag -a v0.9 85fc7e7
$ git log --oneline --decorate --graph
*   88afe0e (HEAD, tag: v1.0, master) Merge branch 'change_site'
|\
| * d7e7346 (change_site) changed the site
* | 14b4dca 新增加一行
|/
* 556f0a0 removed test2.txt
* 2e082b7 add test2.txt
* 048598f add test.txt
* 85fc7e7 (tag: v0.9) test comment from runoob.com
```

如果我们要查看所有标签可以使用以下命令：

```
$ git tag
v0.9
v1.0
```

指定标签信息命令：

```
git tag -a <tagname> -m "runoob.com标签"
```

PGP签名标签命令：

```
git tag -s <tagname> -m "runoob.com标签"
```

← Git 查看提交历史　　Git 远程仓库(Github) →

1 篇笔记

写笔记

1、标签介绍

发布一个版本时，我们通常先在版本库中打一个标签（tag），这样就唯一确定了打标签时刻的版本。将来无论什么时候，取某个标签的版本，就是把那个打标签的时刻的历史版本取出来。

所以，标签也是版本库的一个快照。

Git 的标签虽然是版本库的快照，但其实它就是指向某个 commit 的指针（跟分支很像对不对？但是分支可以移动，标签不能移动），所以，创建和删除标签都是瞬间完成的。

> Git有commit，为什么还要引入tag？
> "请把上周一的那个版本打包发布，commit号是6a5819e..."
> "一串乱七八糟的数字不好找！"
> 如果换一个办法：
> "请把上周一的那个版本打包发布，版本号是v1.2"
> "好的，按照tag v1.2查找commit就行！"
> 所以，tag就是一个让人容易记住的有意义的名字，它跟某个commit绑在一起。

同大多数 VCS 一样，Git 也可以对某一时间点上的版本打上标签。人们在发布某个软件版本（比如 v1.0 等等）的时候，经常这么做。

本节我们一起来学习如何列出所有可用的标签，如何新建标签，以及各种不同类型标签之间的差别。

新建标签

Git 使用的标签有两种类型：**轻量级的（lightweight）**和**含附注的（annotated）**。

轻量级标签就像是个不会变化的分支，实际上它就是个指向特定提交对象的引用。

而含附注标签，实际上是存储在仓库中的一个独立对象，它有自身的校验和信息，包含着标签的名字，电子邮件地址和日期，以及标签说明，标签本身也允许使用 GNU Privacy Guard (GPG) 来签署或验证。

一般我们都建议使用含附注型的标签，以便保留相关信息；

当然，如果只是临时性加注标签，或者不需要旁注额外信息，用轻量级标签也没问题。

2 创建标签

```
[root@Git git]# git tag v1.0
```

3 查看已有标签

```
[root@Git git]# git tag
v1.0
[root@Git git]# git tag v1.1
[root@Git git]# git tag
v1.0
v1.1
```

4 删除标签

```
[root@Git git]# git tag -d v1.1
Deleted tag ‘v1.1’ (was 91388f0)
[root@Git git]# git tag
v1.0
```

5 查看此版本所修改的内容

```
[root@Git git]# git show v1.0
commit 91388f0883903ac9014e006611944f6688170ef4
Author: "syaving" <"819044347@qq.com">
Date: Fri Dec 16 02:32:05 2016 +0800
commit dir
diff –git a/readme b/readme
index 7a3d711..bfecb47 100644
— a/readme
+++ b/readme
@@ -1,2 +1,3 @@
text
hello git
+use commit
[root@Git git]# git log –oneline
91388f0 commit dir
e435fe8 add readme
2525062 add readme
```

宋某人c 2年前 (2017-03-23)