Git submodule 子模块的管理和使用

Maonx 关注 IP属地: 浙江

9 2017.04.16 21:07:42 字数 612 阅读 161,256

使用前提

经常碰到这种情况：当你在一个Git 项目上工作时，你需要在其中使用另外一个Git 项目。也许它是一个第三方开发的Git 库或者是你独立开发和并在多个父项目中使用的。这个情况下一个常见的问题产生了：你想将两个项目单独处理但是又需要在其中一个中使用另外一个。

在Git 中你可以用子模块 `submodule` 来管理这些项目， `submodule` 允许你将一个Git 仓库当作另外一个Git 仓库的子目录。这允许你克隆另外一个仓库到你的项目中并且保持你的提交相对独立。

添加子模块

此文中统一将远程项目 `https://github.com/maonx/vimwiki-assets.git` 克隆到本地 `assets` 文件夹。

```
$ git submodule add https://github.com/maonx/vimwiki-assets.git assets
```

添加子模块后运行 `git status`，可以看到目录有增加1个文件 `.gitmodules`，这个文件用来保存子模块的信息。

```
$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   .gitmodules
    new file:   assets
```

查看子模块

```
$ git submodule
 e33f854d3f51f5ebd771a68da05ad0371a3c0570 assets (heads/master)
```

更新子模块

- 更新项目内子模块到最新版本

```
$ git submodule update
```

- 更新子模块为远程项目的最新版本

```
$ git submodule update --remote
```

克隆包含子模块的项目

克隆包含子模块的项目有二种方法：一种是先克隆父项目，再更新子模块；另一种是直接递归克隆整个项目。

克隆父项目，再更新子模块

1. 克隆父项目

```
$ git clone https://github.com/maonx/vimwiki-assets.git assets
```

1. 查看子模块

```
$ git submodule
 -e33f854d3f51f5ebd771a68da05ad0371a3c0570 assets
```

子模块前面有一个 - ，说明子模块文件还未检入（空文件夹）。

1. 初始化子模块

```
$ git submodule init
Submodule 'assets' (https://github.com/maonx/vimwiki-assets.git) registered for path 'assets'
```

初始化模块只需在克隆父项目后运行一次。

1. 更新子模块

```
$ git submodule update
Cloning into 'assets'...
remote: Counting objects: 151, done.
remote: Compressing objects: 100% (80/80), done.
remote: Total 151 (delta 18), reused 0 (delta 0), pack-reused 70
Receiving objects: 100% (151/151), 1.34 MiB | 569.00 KiB/s, done.
Resolving deltas: 100% (36/36), done.
Checking connectivity... done.
Submodule path 'assets': checked out 'e33f854d3f51f5ebd771a68da05ad0371a3c0570'
```

递归克隆整个项目

```
git clone https://github.com/maonx/vimwiki-assets.git assets --recursive
```

递归克隆整个项目，子模块已经同时更新了，一步到位。

修改子模块

在子模块中修改文件后，直接提交到远程项目分支。

```
$ git add .
$ git ci -m "commit"
$ git push origin HEAD:master
```

删除子模块

删除子模块比较麻烦，需要手动删除相关的文件，否则在添加子模块时有可能出现错误
同样以删除 `assets` 文件夹为例

1. 删除子模块文件夹

```
$ git rm --cached assets
$ rm -rf assets
```

1. 删除 `.gitmodules` 文件中相关子模块信息

```
[submodule "assets"]
  path = assets
  url = https://github.com/maonx/vimwiki-assets.git
```

1. 删除 `.git/config` 中的相关子模块信息

```
[submodule "assets"]
  url = https://github.com/maonx/vimwiki-assets.git
```

1. 删除 `.git` 文件夹中的相关子模块文件

```
$ rm -rf .git/modules/assets
```

最后编辑于： 2017.12.06 19:03:17