

`git fetch` 命令用于从另一个存储库下载对象和引用。

## 使用语法

```
git fetch [<options>] [<repository> [<refspec>...]]
git fetch [<options>] <group>
git fetch --multiple [<options>] [(<repository> | <group>)...]
git fetch --all [<options>]
```

## 描述

从一个或多个其他存储库中获取分支和/或标签(统称为“引用”)以及完成其历史所必需的对象。远程跟踪分支已更新(Git 术语叫做 `commit`)，需要将这些更新取回本地，这时就要用到 `git fetch` 命令。

默认情况下，还会获取指向正在获取的历史记录的任何标签；效果是获取指向您感兴趣的分支的标签。可以使用 `--tags` 或 `--no-tags` 选项或通过配置远程 `.<name>.tagOpt` 来更改此默认行为。通过使用显式提取标签的 `refspec`，可以获取不指向您感兴趣的分支的标签。

`git fetch` 可以从单个命名的存储库或 URL 中获取，也可以从多个存储库中获取，如果给定了 `<group>`，并且配置文件中有一个远程 `<group>` 条目。

获取的参考名称以及它们所指向的对象名称被写入到 `.git/FETCH_HEAD` 中。此信息可能由脚本或其他 git 命令使用，如 `git-pull`。

## 示例

以下是一些示例 -

### 1.更新远程跟踪分支

```
$ git fetch origin
```

上述命令从远程 `refs/heads/` 命名空间复制所有分支，并将它们存储到本地的 `refs/remotes/ origin/` 命名空间中，除非使用分支 `.<name>.fetch` 选项来指定非默认的 `refspec`。

### 2.明确使用 refspec

```
$ git fetch origin +pu:pu maint:tmp
```

此更新(或根据需要创建)通过从远程存储库的分支(分别) `pu` 和 `maint` 提取来分支本地存储库中的 `pu` 和 `tmp`。

即使没有快进，`pu` 分支将被更新，因为它的前缀是加号；`tmp` 不会。

### 3.在远程分支上窥视，无需在本地存储库中配置远程

```
$ git fetch git://git.kernel.org/pub/scm/git/git.git maint
$ git log FETCH_HEAD
```

第一个命令从 `git://git.kernel.org/pub/scm/git/git.git` 从存储库中获取 `maint` 分支，第二个命令使用 `FETCH_HEAD` 来检查具有 `git-log` 的分支。

#### 4.将某个远程主机的更新

```
$ git fetch <远程主机名>
```

要更新所有分支，命令可以简写为：

```
$ git fetch
```

上面命令将某个远程主机的更新，全部取回本地。默认情况下，`git fetch` 取回所有分支的更新。如果只想取回特定分支的更新，可以指定分支名,如下所示 -

```
$ git fetch <远程主机名> <分支名>
```

比如，取回 `origin` 主机的 `master` 分支。

```
$ git fetch origin master
```

所取回的更新，在本地主机上要用“远程主机名/分支名”的形式读取。比如 `origin` 主机的 `master` 分支，就可以用 `origin/master` 读取。

`git branch` 命令的 `-r` 选项，可以用来查看远程分支，`-a` 选项查看所有分支。

```
$ git branch -r
```

```
origin/master
```

```
$ git branch -a
```

```
* master
```

```
remotes/origin/master
```

上面命令表示，本地主机的当前分支是 `master`，远程分支是 `origin/master`。

取回远程主机的更新以后，可以在它的基础上，使用 `git checkout` 命令创建一个新的分支。

```
$ git checkout -b newBrach origin/master
```

上面命令表示，在 `origin/master` 的基础上，创建一个新分支:`newBrach`。

此外，也可以使用 `git merge` 命令或者 `git rebase` 命令，在本地分支上合并远程分支。

```
$ git merge origin/master
```

```
# 或者
```

```
$ git rebase origin/master
```

上面命令表示在当前分支上，合并 `origin/master`。