

`git status` 命令用于显示工作目录和暂存区的状态。使用此命令能看到那些修改被暂存到了, 哪些没有, 哪些文件没有被 Git tracked 到。`git status` 不显示已经 `commit` 到项目历史中去的信息。看项目历史的信息要使用 `git log`。

## 简介

```
git status [<options>...] [--] [<pathspec>...]
```

Shell

## 描述

显示索引文件和当前 HEAD 提交之间的差异, 在工作树和索引文件之间有差异的路径以及工作树中没有被 Git 跟踪的路径。第一个是通过运行 `git commit` 来提交的; 第二个和第三个是你可以通过在运行 `git commit` 之前运行 `git add` 来提交的。

`git status` 相对来说是一个简单的命令, 它简单的展示状态信息。输出的内容分为 3 个分类/组。

```
# On branch master
# Changes to be committed: (已经在 stage 区, 等待添加到 HEAD 中的文件)
# (use "git reset HEAD <file>..." to unstage)
#
#modified: hello.py
#
# Changes not staged for commit: (有修改, 但是没有被添加到 stage 区的文件)
# (use "git add <file>..." to update what will be committed)
# (use "git checkout -- <file>..." to discard changes in working directory)
#
#modified: main.py
#
# Untracked files:(没有 tracked 过的文件, 即从没有 add 过的文件)
# (use "git add <file>..." to include in what will be committed)
#
#hello.pyc
```

Shell

## 忽略文件(untracked 文件)

没有 `tracked` 的文件分为两类. 一是已经被放在工作目录下但是还没有执行 `git add` 的, 另一类是一些编译了的程序文件(如 `.pyc`, `.obj`, `.exe` 等)。当这些不想 `add` 的文件一多起来, `git status` 的输出简直没法看, 一大堆的状态信息怎么看?

基于这个原因。Git 让我们能在一个特殊的文件 `.gitignore` 中把要忽略的文件放在其中, 每一个想忽略的文件应该独占一行, `*` 这个符号可以作为通配符使用。例如在项目根目录下的 `.gitignore` 文件中加入下面内容能阻止 `.pyc` 和 `.tmp` 文件出现在 `git status` 中:

```
*.pyc
*.tmp
```

Shell

## 示例

以下是一些示例 -

在每次执行 `git commit` 之前先使用 `git status` 检查文件状态是一个很好的习惯, 这样能防止你不小心提交了您不想提交的东西。下面的例子展示 `stage` 前后的状态, 并最后提交一个快照.

```
# Edit hello.py
$ git status
# hello.py is listed under "Changes not staged for commit"
$ git add hello.py
$ git status
# hello.py is listed under "Changes to be committed"
$ git commit
$ git status
# nothing to commit (working directory clean)
```

Shell

第一个状态输出显示了这个文件没有被放到暂存区(staged)。`git add` 将影响第二个 `git status` 的输出, 最后一个 `git status` 告诉我们没有什么能可以提交了, 工作目录已经和最近的提交相匹配了。有些命令 (如, `git merge`) 要求工作目录是 `clean` 状态, 这样就不会不小心覆盖更新了。

`git status` 命令可以列出当前目录所有还没有被 git 管理的文件和被 git 管理且被修改但还未提交(`git commit`)的文件。下面来看看如下一个示例 -

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   2.txt
#
```

```
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   1.txt
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       1.log
```

Shell

上面输出结果中“*Changes to be committed*”中所列的内容是在索引中的内容，提交之后进入 Git 工作目录。

上面输出结果中“*Changed but not updated*”中所列的内容是在工作目录中的内容，`git add` 之后将添加进入索引。

上面输出结果中“*Untracked files*”中所列的内容是尚未被 Git 跟踪的内容，`git add` 之后进入添加进入索引。

通过 `git status -uno` 可以只列出所有已经被 git 管理的且被修改但没提交的文件。