**What you need to know about gitignore pattern**

Mon, Sep 7, 2015

After using git for a long time, I thought I knew the basic of gitignore well enough, until I ran into problem with my files not getting committed. There were some basic gitignore patterns that most git users are just not aware of. So some of us might have put something on gitignore without realizing that it does ignore a lot more stuff than we intend.

**The problem I ran into**

I had just moved a directory to another directory.

/some/path/doc # old location

/some/path/nested/doc # new location

Git detects that I’m deleting a bunch of files, but it doesn’t show that I’m adding any new files. I thought git just doesn’t immediately detect the new files. So I committed the changes hoping that now git will detect the new files. It still doesn’t. I wonder what happens if I just move the directory back to its original location. Technically it should revert my changes, right? But moving it back doesn’t help either, git still does not detect that there is any changes in my repository.

I start to suspect gitignore settings. I noticed that at some point someone added this line in gitignore

doc

tmp

That looks suspicious. I thought that just means to ignore any directory named “doc” in the root of the project. But maybe I’m wrong. So I commented out that line to see if it did anything. And it did detect that there were changes to be committed.

When you specify just a word **without leading** or **trailing** slash, it will match any **directory** or **files** with that name anywhere in your project. So having “doc” in .gitignore, means:

# .gitignore

doc

will ignore all of these:

/doc # directory

/doc # file

/some/path/doc # directory

/some/path/doc # file

**What if you just want to ignore the directory at the root level?**

The correct syntax for that is to put leading and trailing slash:

# .gitignore

/doc/

# will ignore

/doc # directory

# will NOT ignore

/doc # file

/some/doc # directory

/another/doc # file

Why the need for trailing slash? If you just put leading slash, it will still ignore the doc **directory** at the root level, but it will also ignore a **file** named “doc” at the root level, if you have a file instead of a directory named “doc” at the root level.

# .gitignore

/doc

will ignore

/doc # directory

/doc # file

So what if we only include trailing slash?

# .gitignore

**doc/**

it will ignore all directory named “doc” anywhere in your project, but it **will not ignore** any **files** named “doc”

# will ignore:

/doc # directory

/some/path/doc # directory

# will NOT ignore:

/another/path/doc # file

**Summary**

* To ignore just directories, add a slash at the end of the pattern. Otherwise it will ignore both files and directories.
* To ignore just files/directories at the root, add a slash at the beginning of the pattern. Otherwise it will ignore files/directories anywhere in your project.

If you’re curious, check out a more detailed documentation about [gitignore patterns](https://git-scm.com/docs/gitignore). These 2 basic patterns seem like a must know for most git users, so you don’t accidentally ignore more stuff than you intended to.

<https://www.hilman.io/blog/2015/09/what-you-need-to-know-about-gitignore/>