

Dealing with special characters in branch and tag names

In this article

About branch and tag names

Why you need to escape special characters

How to escape special characters in branch and tag names

Naming branches and tags

Restrictions on names in GitHub

Git is very permissive about what characters are allowed in branch and tag names. When using Git from a command-line shell, you may need to escape or quote special characters.

About branch and tag names

Most repositories use simple branch names, such as `main` or `update-icons`. Tag names also usually follow a basic format, such as a version number like `v1.2.3`. Both branch names and tag names may also use the path separator (`/`) for structure, for example `area/item` or `level-1/level-2/level-3`. Other than some exceptions — such as not starting or ending a name with a slash, or having consecutive slashes in the name — Git has very few restrictions on what characters may be used in branch and tag names. For more information, see "[git-check-ref-format](#)" in the Git documentation.

Why you need to escape special characters

When using a CLI, you might have situations where a branch or tag name contains special characters that have a special meaning for your shell environment. To use these characters safely in a Git command, they must be quoted or escaped, otherwise the command may have unintended effects.

For example, the `$` character is used by many shells to refer to a variable. Most shells would interpret a valid branch name like `hello-$USER` as equivalent to the word "hello", followed by a hyphen, followed by the current value of the `USER` variable, rather than the literal string `hello-$USER`. If a branch name includes the `$` character, then the shell must be stopped from expanding it as a variable reference. Similarly, if a branch name contains a semi-colon (`;`), most shells interpret it as a command separator, so it needs to be quoted or escaped.

How to escape special characters in branch and tag names

Most branch and tag names with special characters can be handled by including the name in single quotes, for example `'hello-$USER'`.

- In the [Bash](#) shell, enclosing a string of characters in single quotes preserves the

literal value of the characters within the single quotes.

- [Zsh](#) behaves similar to Bash, however this behavior is configurable using the `RC_QUOTES` option.
- [PowerShell](#) also treats characters literally when inside single quotes.

For these shells, the main exception is when the branch or tag name itself contains a single quote. In this case, you should consult the official documentation for your shell:

- [Bash documentation](#)
- [Zsh documentation](#)
- [Fish documentation](#)
- [PowerShell documentation](#)

Naming branches and tags [↗](#)

If possible, create branch and tag names that don't contain special characters, as these would need to be escaped. A safe default set of characters to use for branch names and tag names is:

- The English alphabet (`a` to `z` and `A` to `Z`)
- Numbers (`0` to `9`)
- A limited set of punctuation characters:
 - period (`.`)
 - hyphen (`-`)
 - underscore (`_`)
 - forward slash (`/`)

To avoid confusion, you should start branch names with a letter.

Restrictions on names in GitHub [↗](#)

GitHub restricts a small number of branch and tag names from being pushed up. Those restrictions are:

- No names which look like Git object IDs (40 characters containing only 0-9 and A-F), to prevent confusion with actual Git object IDs.
- No names beginning with `refs/`, to prevent confusion with the full name of Git refs. For more information about refs, see "[Git References](#)" in the Git documentation.

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)