

Git branch name

Valid Git branch name — alphanumeric and limited symbols.

```
^(?!\\|.|.*([\\/]\\.|\\|\\|@\\{|\\|\\|)) [^\\040\\177
~^: ?*\\[ ]+\\/? (?<!\\.lock|[\\/.])$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
(?!)	start of a negative lookahead — NOT followed by
\\	literal “/”
	OR — try the alternative
.*	zero or more of any character
(	start of a capturing group
[\\]	any of: “/”
\\.	literal “.”
	OR — try the alternative
\\	literal “/”
\\	literal “/”
	OR — try the alternative
@	literal text “@”
\\{	literal “{”
	OR — try the alternative
\\	literal “\”
)	end of group
)	end of group
[^\\040\\177 ~^: ?*\\[]+	one or more: any character except “0”, “4”, “1”, “7”, “ ”, “~”, “^”, “.”, “?”, “*”, “[”
\\/?	optional (zero or one): literal “/”
(?<!)	start of a negative lookbehind — NOT preceded by
\\.	literal “.”
lock	literal text “lock”
	OR — try the alternative
[\\/.]	any of: “/”, “.”
)	end of group
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^(?!\\\/ .*([/\]\.\ \\\\\/ @{\ \\\}))[^\040\\177 ~^:?*\[+\\\/?(?!\.lock [\/\.]\$)/; re.test(value);</pre>
Python	<pre>import re re.match(r"^(!\\/ . *([\\/]\. \\/\/ @{ \\)) [^\040\\177 ~^:?*\[+\\\/?(?!\\.lock [\/.])\$", value)</pre>
Java	<pre>Pattern.compile("^(!\\/ . *([\\/]\. \\/\/ @{ \\)) [^\040\\177 ~^:?*\[+\\\/?(?!\.lock [\/.])\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^(?!\\/ . *([\\/]\. \\/\/ @{ \\)) [^\040\\177 ~^:?*\[+\\\/?(?!\.lock [\/.])\$");</pre>
Go	<pre>regexp.MustCompile(`^(?!\\/ . *([\\/]\.  \\/\/ @{  \\)) [^\040\\177 ~^:?*\[+\\\/?(?!\.lock [\/.])\$`).MatchString(value)</pre>
Ruby	<pre>/^(?!\\/ . *([\\/]\. \\/\/ @{ \\)) [^\040\\177 ~^:?*\[+\\\/?(?!\.lock [\/\.]\$)/.match?(value)</pre>
PHP	<pre>preg_match('~^(?!\\/ . *([\\/]\. \\/\/ @{ \\)) [^\040\\177 \n~^:?*\[+\\\/?(?!\.lock [\/.])\$~', \$value);</pre>

Common pitfalls

- **Anchored to the whole string.** This pattern uses `^` and `$`, so it requires the entire input to match. To find it inside a longer text, drop the anchors and use the global (`g`) flag.
- **Greedy “`.*`”.** A greedy `.* / .+` can match more than intended. Use a lazy version (`.*?`) or a negated class (`[^...]`) to stop at the right place.
- **Escape it correctly per language.** In Java and JavaScript strings each backslash must be doubled (`\\d`); in Python, Go, and C# use raw/verbatim strings so the backslashes survive.
- **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the git branch name against a source of truth (database, API, or checksum) where it matters.