

URL / web address

HTTP or HTTPS URL with optional path, query, and fragment.

```
^https?:\/\/[^\s/$.?\#] . [^\s]*$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
http	literal text "http"
s?	optional (zero or one): the literal "s"
://	literal text "://"
[^\s/\$.?\#]	any character except whitespace, "/", "\$", ".", "?", "#"
.	any character (except newline)
[^\s]*	zero or more: any character except whitespace
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^https?:\/\/[^\s/\$.?\#] . [^\s]*\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^https?:\/\/[^\s/\$.?\#] . [^\s]*\$", value)</pre>
Java	<pre>Pattern.compile("^https?:\/\/[^\s/\$.?\#] . [^\s]*\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^https?:\/\/[^\s/\$.?\#] . [^\s]*\$");</pre>
Go	<pre>regexp.MustCompile(`^https?:\/\/[^\s/\$.?\#] . [^\s]*\$`).MatchString(value)</pre>
Ruby	<pre>/^https?:\/\/[^\s/\$.?\#] . [^\s]*\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^https?:\/\/[^\s/\$.?\#] . [^\s]*\$~', \$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.
- **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 url / web address against a source of truth (database, API, or checksum) where it matters.