

HTTP header line

Header-Name: value format.

```
^([A-Za-z][A-Za-z0-9-]*):\s*(.+)\s*$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
(	start of a capturing group
[A-Za-z]	any of: uppercase letters, lowercase letters
[A-Za-z0-9-]*	zero or more: any of: uppercase letters, lowercase letters, digits, "-"
)	end of group
:	literal text ":"
\s*	zero or more: whitespace
(	start of a capturing group
.+?	one or more (lazy — as few as possible) of any character
)	end of group
\s*	zero or more: whitespace
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^([A-Za-z][A-Za-z0-9-]*):\s*(.+)\s*\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^([A-Za-z][A-Za-z0-9-]*):\s*(.+)\s*\$", value)</pre>
Java	<pre>Pattern.compile("^([A-Za-z][A-Za-z0-9-]*):\\s*(.+)?\\s*\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^([A-Za-z][A-Za-z0-9-]*):\s*(.+)\s*\$");</pre>
Go	<pre>regexp.MustCompile(`^([A-Za-z][A-Za-z0-9-]*):\s*(.+)\s*\$`).MatchString(value)</pre>
Ruby	<pre>/^([A-Za-z][A-Za-z0-9-]*):\s*(.+)\s*\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^([A-Za-z][A-Za-z0-9-]*):\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.
- **ASCII letters only.** [a-zA-Z] excludes accented and non-Latin letters (é, ü, ß, ñ, and non-Latin scripts). For international input use Unicode properties like \p{L} with the u 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 http header line against a source of truth (database, API, or checksum) where it matters.