

24-hour time (HH:MM)

Time in 24-hour format with optional seconds.

```
^([01]\d|2[0-3]):[0-5]\d(:[0-5]\d)?$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
(	start of a capturing group
[01]	any of: "0", "1"
\d	any digit (0–9)
	OR — try the alternative
2	literal text "2"
[0-3]	any of: 0–3
)	end of group
:	literal text ":"
[0-5]	any of: 0–5
\d	any digit (0–9)
(	start of a capturing group
:	literal text ":"
[0-5]	any of: 0–5
\d	any digit (0–9)
)?	end of group, optional (zero or one)
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^([01]\d 2[0-3]):[0-5]\d(:[0-5]\d)?\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^([01]\d 2[0-3]):[0-5]\d(:[0-5]\d)?\$", value)</pre>
Java	<pre>Pattern.compile("^([01]\\d 2[0-3]):[0-5]\\d(:[0-5]\\d)?\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^([01]\d 2[0-3]):[0-5]\d(:[0-5]\d)?\$");</pre>
Go	<pre>regexp.MustCompile(`^([01]\d 2[0-3]):[0-5]\d(:[0-5]\d)?\$`).MatchString(value)</pre>
Ruby	<pre>/^([01]\d 2[0-3]):[0-5]\d(:[0-5]\d)?\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^([01]\d 2[0-3]):[0-5]\d(:[0-5]\d)?\$~', \$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 24-hour time (`hh:mm`) against a source of truth (database, API, or checksum) where it matters.