

RFC 2822 date

Email-style date format like RFC 2822.

```
^(Mon|Tue|Wed|Thu|Fri|Sat|Sun), \d{1,2}  
(Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec) \d{4}  
\d{2}:\d{2}:\d{2} [+ -]\d{4}$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
(	start of a capturing group
Mon	literal text "Mon"
	OR — try the alternative
Tue	literal text "Tue"
	OR — try the alternative
Wed	literal text "Wed"
	OR — try the alternative
Thu	literal text "Thu"
	OR — try the alternative
Fri	literal text "Fri"
	OR — try the alternative
Sat	literal text "Sat"
	OR — try the alternative
Sun	literal text "Sun"
)	end of group
,	literal text " , "
\d{1,2}	between 1 and 2 times: digit (0–9)
" "	literal text " "
(	start of a capturing group
Jan	literal text "Jan"
	OR — try the alternative
Feb	literal text "Feb"
	OR — try the alternative
Mar	literal text "Mar"
	OR — try the alternative
Apr	literal text "Apr"

Token	Meaning
	OR — try the alternative
May	literal text "May"
	OR — try the alternative
Jun	literal text "Jun"
	OR — try the alternative
Jul	literal text "Jul"
	OR — try the alternative
Aug	literal text "Aug"
	OR — try the alternative
Sep	literal text "Sep"
	OR — try the alternative
Oct	literal text "Oct"
	OR — try the alternative
Nov	literal text "Nov"
	OR — try the alternative
Dec	literal text "Dec"
)	end of group
	literal text " "
\d{4}	exactly 4 times: digit (0–9)
	literal text " "
\d{2}	exactly 2 times: digit (0–9)
:	literal text ":"
\d{2}	exactly 2 times: digit (0–9)
:	literal text ":"
\d{2}	exactly 2 times: digit (0–9)
	literal text " "
[+-]	any of: "+", "-"
\d{4}	exactly 4 times: digit (0–9)
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript

```
const re = /^(Mon|Tue|Wed|Thu|Fri|Sat|Sun), \d{1,2}
(Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec) \d{4} \d{2}:\d{2}:\d{2} [+-]\d{4}$;/
re.test(value);
```

Python

```
import re
re.match(r"^(Mon|Tue|Wed|Thu|Fri|Sat|Sun), \d{1,2}
(Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec) \d{4} \d{2}:\d{2}:\d{2} [+-]\d{4}$",
value)
```

Java	<pre>Pattern.compile("(Mon Tue Wed Thu Fri Sat Sun), \\d{1,2} (Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec) \\d{4} \\d{2}:\\d{2}:\\d{2} [+-]\\d{4}\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^(Mon Tue Wed Thu Fri Sat Sun), \d{1,2} (Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec) \d{4} \d{2}:\d{2}:\d{2} [+-]\d{4}\$");</pre>
Go	<pre>regexp.MustCompile(`^(Mon Tue Wed Thu Fri Sat Sun), \d{1,2} (Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec) \d{4} \d{2}:\d{2}:\d{2} [+-]\d{4}\$`).MatchString(value)</pre>
Ruby	<pre>/^(Mon Tue Wed Thu Fri Sat Sun), \d{1,2} (Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov D ec) \d{4} \d{2}:\d{2}:\d{2} [+-]\d{4}\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^(Mon Tue Wed Thu Fri Sat Sun), \d{1,2} (Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec) \d{4} \d{2}:\d{2}:\d{2} [+-]\d{4}\$~', \$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 rfc 2822 date against a source of truth (database, API, or checksum) where it matters.