

International phone (E.164)

The international standard: + and 7–15 digits.

```
^\+[1-9]\d{6,14}$
```

Token-by-token breakdown

Token	Meaning
<code>^</code>	start of string (or line in multiline mode)
<code>\+</code>	literal "+"
<code>[1-9]</code>	any of: 1–9
<code>\d{6,14}</code>	between 6 and 14 times: digit (0–9)
<code>\$</code>	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /\^[1-9]\d{6,14}\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^\+[1-9]\d{6,14}\$", value)</pre>
Java	<pre>Pattern.compile("^\\+[1-9]\\d{6,14}\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^\+[1-9]\d{6,14}\$");</pre>
Go	<pre>regexp.MustCompile(`^\+[1-9]\d{6,14}\$`).MatchString(value)</pre>
Ruby	<pre>/^\+[1-9]\d{6,14}\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^\+[1-9]\d{6,14}\$~', \$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 international phone (e.164) against a source of truth (database, API, or checksum) where it matters.