

GSTIN

15-character Goods and Services Tax Identification Number.

```
^[0-9]{2}[A-Z]{5}[0-9]{4}[A-Z][1-9A-Z]Z[0-9A-Z]$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
[0-9]{2}	exactly 2 times: any of: digits
[A-Z]{5}	exactly 5 times: any of: uppercase letters
[0-9]{4}	exactly 4 times: any of: digits
[A-Z]	any of: uppercase letters
[1-9A-Z]	any of: 1–9, uppercase letters
Z	literal text "Z"
[0-9A-Z]	any of: digits, uppercase letters
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^[0-9]{2}[A-Z]{5}[0-9]{4}[A-Z][1-9A-Z]Z[0-9A-Z]\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^[0-9]{2}[A-Z]{5}[0-9]{4}[A-Z][1-9A-Z]Z[0-9A-Z]\$", value)</pre>
Java	<pre>Pattern.compile("^[0-9]{2}[A-Z]{5}[0-9]{4}[A-Z][1-9A-Z]Z[0-9A-Z]\$").matcher(value).matc hes();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^[0-9]{2}[A-Z]{5}[0-9]{4}[A-Z][1-9A-Z]Z[0-9A-Z]\$");</pre>
Go	<pre>regexp.MustCompile(`^[0-9]{2}[A-Z]{5}[0-9]{4}[A-Z][1-9A-Z]Z[0-9A-Z]\$`).MatchString(valu e)</pre>
Ruby	<pre>/^[0-9]{2}[A-Z]{5}[0-9]{4}[A-Z][1-9A-Z]Z[0-9A-Z]\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^[0-9]{2}[A-Z]{5}[0-9]{4}[A-Z][1-9A-Z]Z[0-9A-Z]\$~', \$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.
- **Uppercase only.** Ranges like [A-Z] won't match lowercase. Add a-z (or the i flag) if lowercase input should be accepted.
- **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the gstin against a source of truth (database, API, or checksum) where it matters.