

US ITIN

Individual Taxpayer ID — like SSN but starts with 9.

```
^9\d{2}-(7\d|8[0-8]|9[0-2]|9[4-9])-\d{4}$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
9	literal text "9"
\d{2}	exactly 2 times: digit (0–9)
-	literal text "-"
(	start of a capturing group
7	literal text "7"
\d	any digit (0–9)
	OR — try the alternative
8	literal text "8"
[0-8]	any of: 0–8
	OR — try the alternative
9	literal text "9"
[0-2]	any of: 0–2
	OR — try the alternative
9	literal text "9"
[4-9]	any of: 4–9
)	end of group
-	literal text "-"
\d{4}	exactly 4 times: digit (0–9)
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^9\d{2}-(7\d 8[0-8] 9[0-2] 9[4-9])-\d{4}\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^9\d{2}-(7\d 8[0-8] 9[0-2] 9[4-9])-\d{4}\$", value)</pre>
Java	<pre>Pattern.compile("^9\\d{2}-(7\\d 8[0-8] 9[0-2] 9[4-9])-\\d{4}\$").matcher(value).matches() ;</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^9\d{2}-(7\d 8[0-8] 9[0-2] 9[4-9])-\d{4}\$");</pre>
Go	<pre>regexp.MustCompile(`^9\d{2}-(7\d 8[0-8] 9[0-2] 9[4-9])-\d{4}\$`).MatchString(value)</pre>

Ruby	<code>/^9\d{2}-(7\d 8[0-8] 9[0-2] 9[4-9])-\d{4}\$/ .match?(value)</code>
PHP	<code>preg_match('^9\d{2}-(7\d 8[0-8] 9[0-2] 9[4-9])-\d{4}\$~', \$value);</code>

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 us itin against a source of truth (database, API, or checksum) where it matters.