REGEX CHEAT SHEET

regexguide.com/patterns/linkedin-profile.html

LinkedIn profile URL

linkedin.com/in/handle URL.

```
^https?:\/\/(www\.)?linkedin\.com\/in\/[A-Za-z0-9\-_%]+\/?$
```

Token-by-token breakdown

Token	Meaning
`^`	start of string (or line in multiline mode)
`http`	literal text “http”
`s?`	optional (zero or one): the literal “s”
`:`	literal text “:”
`\/`	literal “/”
`\/`	literal “/”
`(`	start of a capturing group
`www`	literal text “www”
`\.`	literal “.”
`)?`	end of group, optional (zero or one)
`linkedin`	literal text “linkedin”
`\.`	literal “.”
`com`	literal text “com”
`\/`	literal “/”
`in`	literal text “in”
`\/`	literal “/”
`[A-Za-z0-9\-_%]+`	one or more: any of: uppercase letters, lowercase letters, digits, “-”, “_”, “%”
`\/?`	optional (zero or one): literal “/”
`$`	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript

```
const re = /^https?:\\/\\/(www\.)?linkedin\.com\\/in\\/[A-Za-z0-9\-_%]+\\/?$/;
re.test(value);
```

Python

```
import re
re.match(r"^https?:\/\/(www\.)?linkedin\.com\/in\/[A-Za-z0-9\-_%]+\/?$", value)
```

Java

```
Pattern.compile("^https?:\\/\\/(www\\.)?linkedin\\.com\\/in\\/[A-Za-z0-9\\-_%]+\\/?$").
matcher(value).matches();
```

C# / .NET

```
Regex.IsMatch(value, @"^https?:\/\/(www\.)?linkedin\.com\/in\/[A-Za-z0-9\-_%]+\/?$");
```

Go

```
regexp.MustCompile(`^https?:\/\/(www\.)?linkedin\.com\/in\/[A-Za-z0-9\-_%]+\/?$`).Match
String(value)
```

Ruby	`/^https?:\\/\\/(www\.)?linkedin\.com\\/in\\/[A-Za-z0-9\-_%]+\\/?$/.match?(value)`
PHP	`preg_match('~^https?:\/\/(www\.)?linkedin\.com\/in\/[A-Za-z0-9\-_%]+\/?$~', $value);`

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.
- **ASCII letters only.** [a-zA-Z] excludes accented and non-Latin letters (é, ü, ß, ñ, and non-Latin scripts). For international input use Unicode properties like \p{L} with the u 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 linkedin profile url against a source of truth (database, API, or checksum) where it matters.