

Base64 string

Base64-encoded data with optional padding.

```
^[A-Za-z0-9+/]{0,2}$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
[A-Za-z0-9+/]+	one or more: any of: uppercase letters, lowercase letters, digits, "+", "/"
={0,2}	between 0 and 2 times: the literal "="
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript `const re = /^[A-Za-z0-9+/]{0,2}$/;
re.test(value);`

Python `import re
re.match(r"^[A-Za-z0-9+/]{0,2}$", value)`

Java `Pattern.compile("^[A-Za-z0-9+/]{0,2}$").matcher(value).matches();`

C# / .NET `Regex.IsMatch(value, @"^[A-Za-z0-9+/]{0,2}$");`

Go `regexp.MustCompile(`^[A-Za-z0-9+/]{0,2}$`).MatchString(value)`

Ruby `/^[A-Za-z0-9+/]{0,2}$/ .match?(value)`

PHP `preg_match('~^[A-Za-z0-9+/]{0,2}$~', $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.
- **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the base64 string against a source of truth (database, API, or checksum) where it matters.