RuPay card

RuPay — India's domestic card network (NPCI).

```
^(508[5-9]|6069|607[0-9]|608[0-7]|6521|6522)\d{12}$
```

Token-by-token breakdown

Token	Meaning
`^`	start of string (or line in multiline mode)
`(`	start of a capturing group
`508`	literal text “508”
`[5-9]`	any of: 5–9
`|`	OR — try the alternative
`6069`	literal text “6069”
`|`	OR — try the alternative
`607`	literal text “607”
`[0-9]`	any of: digits
`|`	OR — try the alternative
`608`	literal text “608”
`[0-7]`	any of: 0–7
`|`	OR — try the alternative
`6521`	literal text “6521”
`|`	OR — try the alternative
`6522`	literal text “6522”
`)`	end of group
`\d{12}`	exactly 12 times: digit (0–9)
`$`	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript

```
const re = /^(508[5-9]|6069|607[0-9]|608[0-7]|6521|6522)\d{12}$/;
re.test(value);
```

Python

```
import re
re.match(r"^(508[5-9]|6069|607[0-9]|608[0-7]|6521|6522)\d{12}$", value)
```

Java

```
Pattern.compile("^(508[5-9]|6069|607[0-9]|608[0-7]|6521|6522)\\d{12}$").matcher(value).
matches();
```

C# / .NET

```
Regex.IsMatch(value, @"^(508[5-9]|6069|607[0-9]|608[0-7]|6521|6522)\d{12}$");
```

Go

```
regexp.MustCompile(`^(508[5-9]|6069|607[0-9]|608[0-7]|6521|6522)\d{12}$`).MatchString(v
alue)
```

Ruby	`/^(508[5-9]|6069|607[0-9]|608[0-7]|6521|6522)\d{12}$/.match?(value)`
PHP	`preg_match('~^(508[5-9]|6069|607[0-9]|608[0-7]|6521|6522)\d{12}$~', $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.
- **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 rupay card against a source of truth (database, API, or checksum) where it matters.