# URL query parameter

key=value query parameter.

```
(?:^|[?&])([^=&]+)=([^&]*)
```

## Token-by-token breakdown

| Token | Meaning |
|---|---|
| `(?:` | start of a non-capturing group |
| `^` | start of string (or line in multiline mode) |
| `\|` | OR — try the alternative |
| `[?&]` | any of: “?”, “&” |
| `)` | end of group |
| `(` | start of a capturing group |
| `[^=&]+` | one or more: any character except “=”, “&” |
| `)` | end of group |
| `=` | literal text “=” |
| `(` | start of a capturing group |
| `[^&]*` | zero or more: any character except “&” |
| `)` | end of group |

## Quick usage in 7 languages

**JavaScript**

```
const re = /(?:^|[?&])([^=&]+)=([^&]*)/;
re.test(value);
```

**Python**

```
import re
re.match(r"(?:^|[?&])([^=&]+)=([^&]*)", value)
```

**Java**

```
Pattern.compile("(?:^|[?&])([^=&]+)=([^&]*)").matcher(value).matches();
```

**C# / .NET**

```
Regex.IsMatch(value, @"(?:^|[?&])([^=&]+)=([^&]*)");
```

**Go**

```
regexp.MustCompile(`(?:^|[?&])([^=&]+)=([^&]*)`).MatchString(value)
```

**Ruby**

```
/(?:^|[?&])([^=&]+)=([^&]*)/.match?(value)
```

**PHP**

```
preg_match('~(?:^|[?&])([^=&]+)=([^&]*)~', $value);
```

## Common pitfalls

- **Not anchored.** Without ^ and $ this can match a substring anywhere in the input — add anchors if you need the whole value to conform.
- **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the url query parameter against a source of truth (database, API, or checksum) where it matters.