Traefik ReplacePathRegex Documentation

- Configuration Examples
- Configuration Options

The `replacePathRegex` middleware will:

- Replace the matching path with the specified one.
- Store the original path in an `X-Replaced-Path` header

Configuration Examples

```
# Replace path with regex
http:
  middlewares:
    test-replacepathregex:
      replacePathRegex:
        regex: "^/foo/(.*)"
        replacement: "/bar/$1"
```

```
# Replace path with regex
[http.middlewares]
  [http.middlewares.test-replacepathregex.replacePathRegex]
    regex = "^/foo/(.*)"
    replacement = "/bar/$1"
```

```
# Replace path with regex
apiVersion: traefik.io/v1alpha1
kind: Middleware
metadata:
```

```
  name: test-replacepathregex
spec:
  replacePathRegex:
    regex: "^/foo/(.*)"
    replacement: "/bar/$1"
```

```
# Replace path with regex
labels:
  - "traefik.http.middlewares.test-replacepathregex.replacepathregex.regex=^/foo/(.*)"
  - "traefik.http.middlewares.test-replacepathregex.replacepathregex.replacement=/bar/$$1"
```

```
# Replace path with regex
- "traefik.http.middlewares.test-replacepathregex.replacepathregex.regex=^/foo/(.*)"
- "traefik.http.middlewares.test-replacepathregex.replacepathregex.replacement=/bar/$1"
```

Configuration Options

Field	Description	Default	Required
`regex`	Regular expression to match and capture the path from the request URL.		Yes
`replacement`	Replacement path format, which can include captured variables. `$1x` is equivalent to `${1x}`, not `${1}x` (see Regexp.Expand), so use `${1}` syntax.		No

!!! tip

```
Regular expressions and replacements can be tested using online tools such as [Go
Playground](https://play.golang.org/p/mWU9p-wk2ru) or the
[Regex101](https://regex101.com/r/58sIgx/2).

When defining a regular expression within YAML, any escaped character needs to be escaped
twice: `example\.com` needs to be written as `example\\.com`.
```

□□□ #2
□ □□□ □□□ 2026-10-31 14:03:54 UTC
□ □□□ □□□ 2026-10-31 14:03:54 UTC