regexp, do you need it?

Joey Geiger

Examples

regexp.Match inline

```
func IsAuthorizationRegexp(m string) bool {
    r := regexp.MustCompile(`1::Authorize`)
    return r.Match([]byte(m))
}
```

regexp.Match precompiled

```
var (
    authorizationCompiled = regexp.MustCompile(`^1::Authorize`)
)

func IsAuthorizationRegexpCompiled(m string) bool {
    return authorizationCompiled.Match([]byte(m))
}
```

regexp.MatchString precompiled

```go
var (
    authorizationCompiled = regexp.MustCompile(`^1::Authorize`)
)

func IsAuthorizationRegexpCompiledMatchString(m string) bool {
    return authorizationCompiled.MatchString(m)
}
```

strings.Contains

```
func IsAuthorizationStringsContains(m string) bool {
    return strings.Contains(m, `1::Authorize`)
}
```

strings.Contains with constant

```
const (
    authorize = `1::Authorize`
)

func IsAuthorizationStringsContainsConstant(m string) bool {
    return strings.Contains(m, authorize)
}
```

strings.Prefix

```
func IsAuthorizationStringsPrefix(m string) bool {
    return strings.HasPrefix(m, `1::Authorize`)
}
```

Matching Prefix and Suffix

regexp.Match with expected prefix and suffix

```go
var (
    unitRegex = regexp.MustCompile(`^1.7.4.2.\d+::evData`)
)

func IsUnitMessage(m string) bool {
    return unitRegex.Match([]byte(m))
}
```

strings.HasPrefix and strings.TrimPrefix

```go
const (
    statusIDUnitPrefix = `1.7.4.2.`
    evDataPrefix       = `evData`
)

func IsStatusIDUnitMessage(m string) bool {
    if !strings.HasPrefix(m, statusIDUnitPrefix) {
        return false
    }
    s := strings.TrimPrefix(m, statusIDUnitPrefix)
    r := strings.Split(s, "::")
    if len(r) < 2 {
        return false
    }
    return strings.HasPrefix(r[1], evDataPrefix)
}
```

Benchmark

regexp.Match inline

```
func IsAuthorizationRegexp(m string) bool {
    r := regexp.MustCompile(`1::Authorize`)
    return r.Match([]byte(m))
}
```

Thank you

Joey Geiger

Software Engineer at Nexia

@jgeiger

github.com/jgeiger

joeygeiger.com

www.nexiahome.com