



链滴

golang json.Marshal 特殊 html 字符被转义解决方案

作者: [450370050](#)

原文链接: <https://ld246.com/article/1524558037151>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

go语言提供了json的编解码包, json字符串作为参数值传输时发现, json.Marshal生成json特殊字符、>、&会被转义。

```
type Test struct {
    Content    string
}
func main() {
    t := new(Test)
    t.Content = "http://www.baidu.com?id=123&test=1"
    jsonByte, _ := json.Marshal(t)
    fmt.Println(string(jsonByte))
}

{"Content":"http://www.baidu.com?id=123\u0026test=1"}
Process finished with exit code 0
```

GoDoc描述

String values encode as JSON strings coerced to valid UTF-8, replacing invalid bytes with the Unicode replacement rune.

The angle brackets "<" and ">" are escaped to "\u003c" and "\u003e" to keep some browsers from misinterpreting JSON output as HTML.

Ampersand "&" is also escaped to "\u0026" for the same reason.

This escaping can be disabled using an Encoder that had SetEscapeHTML(false) called on it.

json.Marshal 默认 escapeHtml 为true,会转义 <、>、&

```
func Marshal(v interface{}) ([]byte, error) {
    e := &encodeState{}
    err := e.marshal(v, encOpts{escapeHTML: true})
    if err != nil {
        return nil, err
    }
    return e.Bytes(), nil
}
```

解决方案

方法一:

```
content = strings.Replace(content, "\\u003c", "<", -1)
content = strings.Replace(content, "\\u003e", ">", -1)
content = strings.Replace(content, "\\u0026", "&", -1)
```

这种方式比较直接, 硬性字符串替换。比较憨厚

方法二:

文档中写到 ``This escaping can be disabled using an Encoder that had `SetEscapeHTML(false)` called on it. ``

1. 我们先创建一个buffer用于存储json
2. 创建一个jsonencoder
3. 设置html编码为false

```
type Test struct {
    Content    string
}
func main() {
    t := new(Test)
    t.Content = "http://www.baidu.com?id=123&test=1"
    bf := bytes.NewBuffer([]byte{})
    jsonEncoder := json.NewEncoder(bf)
    jsonEncoder.SetEscapeHTML(false)
    jsonEncoder.Encode(t)
    fmt.Println(bf.String())
}
```

```
{"Content": "http://www.baidu.com?id=123&test=1"}
Process finished with exit code 0
```

查看文档和源码还是解决问题的好方法