Golang 练习笔记01

2020/07/10

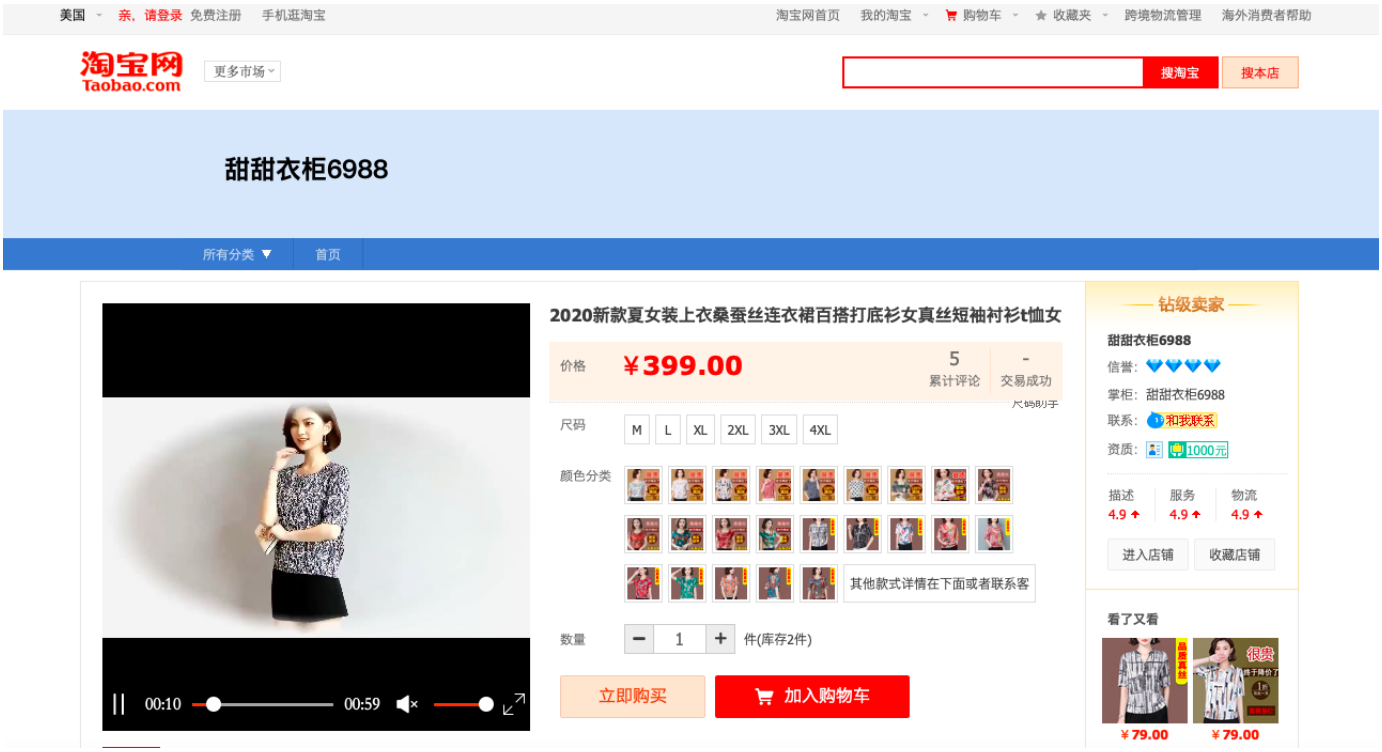
cyc/Golang 练习笔记/Golang 练习笔记01

使用Golang爬取网站数据

目标

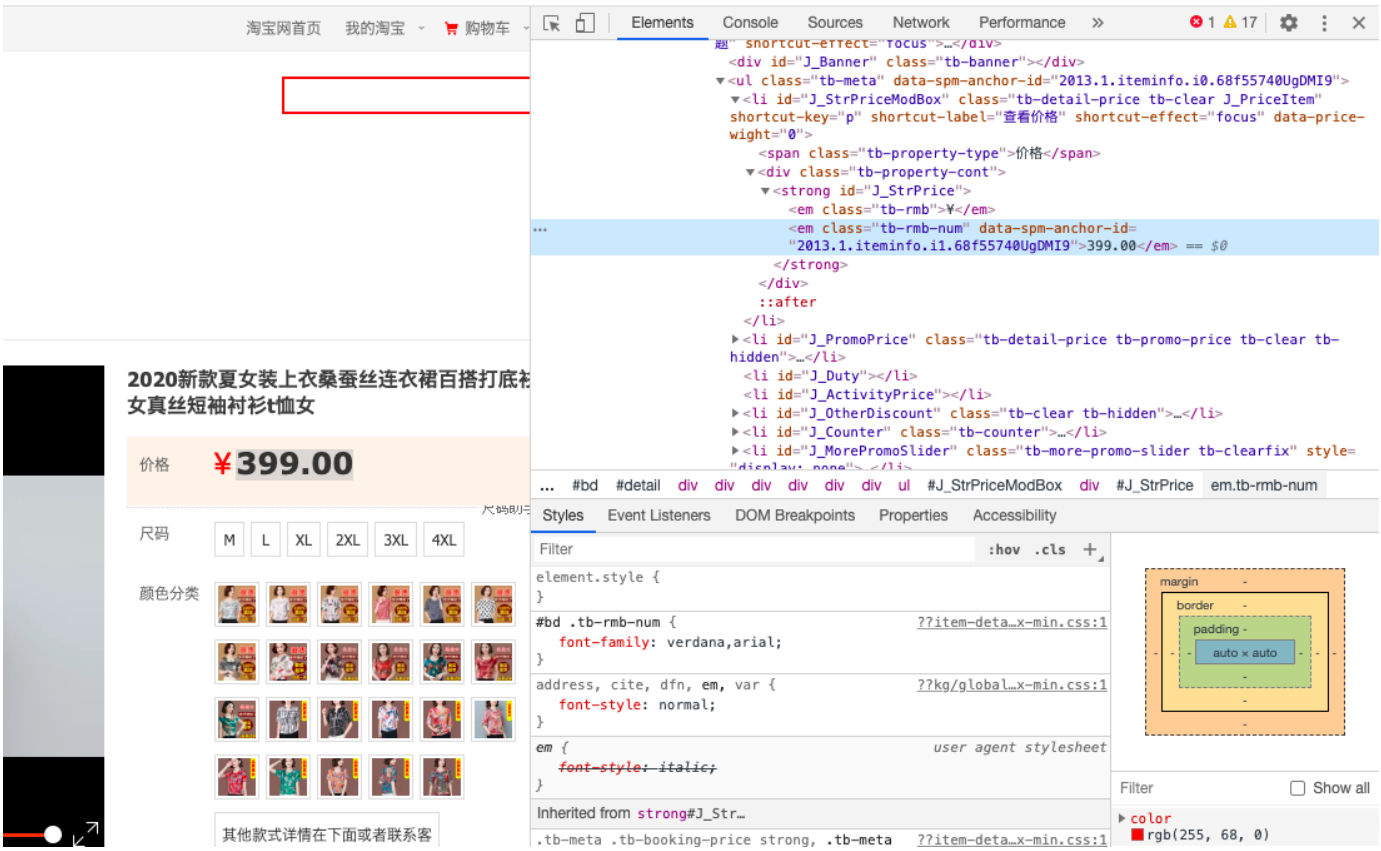
使用Golang爬取一个淘宝网站上的价格信息

https://item.taobao.com/item.htm?id=622329071715



网页中的元素

在Google Chrome中可使用**alt+command+i**来查看网页元素



工具

- 使用colly来爬取数据
- 使用htmlquery来帮助解析数据

代码

example.go

```
package main

import (
    "bytes"
    "fmt"
    "os"
    "strings"

    "github.com/antchfx/htmlquery"
    "github.com/gocolly/colly"
    "golang.org/x/net/html"
)

func main() {
    // Instantiate default collector
    c := colly.NewCollector(
        colly.MaxDepth(1),
    )

    c.OnResponse(func(r *colly.Response) {
        // 以下代码将打印得到的response body的全部内容
        // fmt.Println("body", string(r.Body))
        // 解析response body
        doc, err := htmlquery.Parse(strings.NewReader(string(r.Bo
dy)))
        if err != nil {
            fmt.Fprintf(os.Stderr, "Error message: %v\n", err)
            os.Exit(1)
        } // if
        // 用htmlquery.Find寻找class="tb-rmb-num"的em
        list := htmlquery.Find(doc, "//em[@class=\"tb-rmb-num\"]"
```

```
    )
31          // 使用htmlquery.InnerText()获取内容
32          // htmlquery.Innertext()输入一个*html.Node, 返回一个string
33          fmt.Println("The price is: " + htmlquery.InnerText(list[0
    ]))
34          // 以下代码将html.Node转化为string
35          var b bytes.Buffer
36          errR := html.Render(&b, list[0])
37          if errR != nil {
38              fmt.Fprintf(os.Stderr, "Error message: %v\n", errR)
39              os.Exit(1)
40          } // if
41          fmt.Println(b.String())
42      })
43
44      c.Visit("https://item.taobao.com/item.htm?id=622329071715")
45  } // main
```

运行结果

```
1 $ go run example.go
  [ruby-2.6.3p62]
2 The price is: 399.00
3 <em class="tb-rmb-num">399.00</em>
```

代码在GitHub的practice-go-project repository中的practice-go-project目录下可见
https://github.com/YechengChu/practice-go-project/tree/master/proj0

参考资料

colly
https://github.com/gocolly/colly
https://pkg.go.dev/github.com/gocolly/colly/v2?tab=doc
http://go-colly.org
htmlquery
https://github.com/antchfx/htmlquery

https://godoc.org/github.com/antchfx/htmlquery

Xpath语法详解

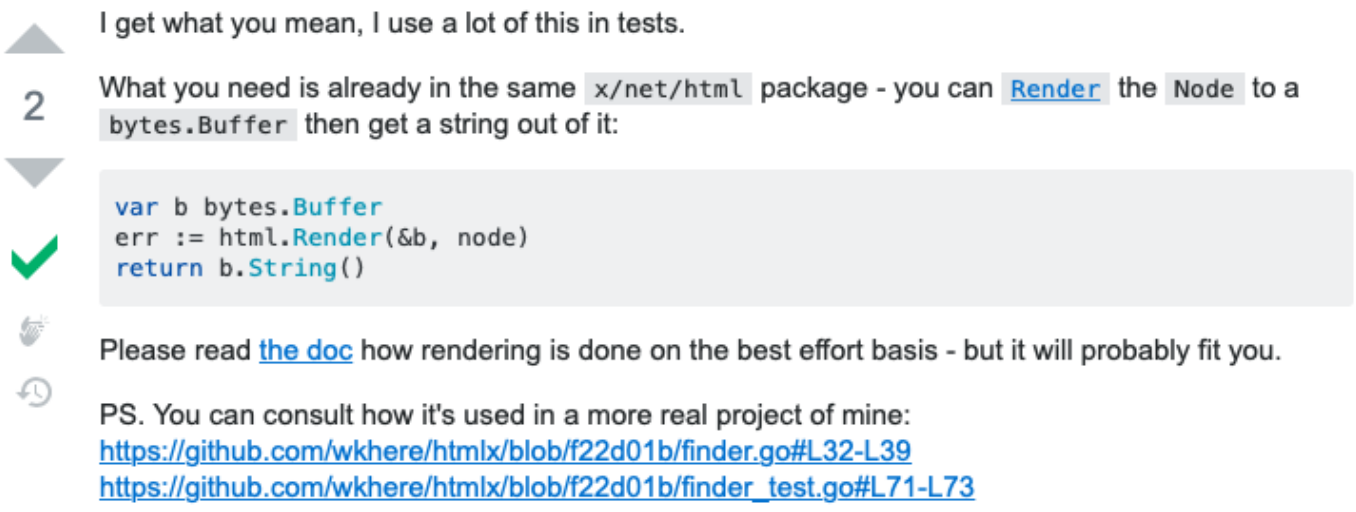
https://www.cnblogs.com/xufengnian/p/10788195.html

菜鸟教程——XPath语法

https://www.runoob.com/xpath/xpath-syntax.html

Retrieve raw data from html.Node

https://stackoverflow.com/questions/48611185/retrieve-raw-data-from-html-node



2

I get what you mean, I use a lot of this in tests.

What you need is already in the same `x/net/html` package - you can `Render` the `Node` to a `bytes.Buffer` then get a string out of it:

```
var b bytes.Buffer
err := html.Render(&b, node)
return b.String()
```

Please read the doc how rendering is done on the best effort basis - but it will probably fit you.

PS. You can consult how it's used in a more real project of mine:
https://github.com/wkhere/htmlx/blob/f22d01b/finder.go#L32-L39
https://github.com/wkhere/htmlx/blob/f22d01b/finder_test.go#L71-L73