

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Московский государственный технический университет  
имени Н.Э. Баумана»  
(МГТУ им. Н.Э. Баумана)

Факультет: Информатика и системы управления  
Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа №2  
«Клиент и сервер HTTP»  
по курсу: «Компьютерные сети»

Выполнил:  
Студент группы ИУ9-31Б  
Гречко Г.В.

Проверил:  
Посевин Д.П.

Москва, 2022

# Цели

Целью данной работы является создание HTTP-клиента и HTTP-сервера на языке Go.

## Задачи

В ходе выполнения лабораторной работы нужно разработать на языке Go HTTP-сервер, который формирует динамические HTML-страницы на основе данных, получаемых с указанного web-сервера. Детали задания приведены в таблицах ниже. Основные требования к HTTP-серверу:

1. полная проверка данных, получаемых из сети (как от клиента, так и от web-сервера);
2. устойчивость к обрыву соединения;
3. возможность одновременного подключения нескольких клиентов к одному серверу;
4. сервер должен вести подробный лог всех ошибок, а также других важных событий (установка и завершение соединения с клиентом, приём и передача сообщений, и т. п.);
5. в зависимости от варианта список новостей должен быть стилизован и содержать, в случае наличия на сайте-доноре, ссылки на первоисточник открывающейся в новом окне, в случае наличия превью изображения к новости на сайте доноре, то это изображение должно отображаться на реализовываемом HTTP-сервере.

## Решение

### Исходный код

server.go

```
1 package main
2
3 import (
4     "html/template"
5     "net/http"
6
7     log "github.com/mgutz/logxi/v1"
8 )
9
10 const INDEX_HTML = `
11     <!doctype html>
12     <html lang="ru">
13         <head>
14             <meta charset="utf-8">
15             <title>Последние новости с news.com.au</title>
16             <link
17
18 ↪ href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
19             rel="stylesheet"
20             integrity="sha384-
21 ↪ EVSTQN3/azprG1Anm3QDgpJLIIm9Nao0Yz1ztcQTWfSpd3yD65VohhpUuCOmLASjC"
22             crossorigin="anonymous"
23             />
24         </head>
25         <body class="bg-dark">
26             <div class="fs-1 fw-bold text-danger text-center mx-3">
27 ↪ ГОРЯЧИЕ НОВОСТИ !!!! </div>
28             <div class="row">
29                 <div class="col-sm">
```

```

27         </div>
28         <div class="col-sm text-center fs-2 py-5">
29             {{if .}}
30                 {{range .}}
31                     <div class="mb-4">
32                         <a href="{{.Ref}}" class="btn btn-secondary
↪ text-center mx-5">{{.Title}}</a>
33                         </img>
34                         <br/>
35                     </div>
36                 {{end}}
37             {{else}}
38                 Не удалось загрузить новости!
39             {{end}}
40         </div>
41         <div class="col-sm">
42         </div>
43     </div>
44 </body>
45 </html>
46 `
47
48 var indexHtml = template.Must(template.New("index").Parse(INDEX_HTML))
49
50 func serveClient(response http.ResponseWriter, request *http.Request) {
51     path := request.URL.Path
52     log.Info("got request", "Method", request.Method, "Path", path)
53     if path != "/" && path != "/index.html" {
54         log.Error("invalid path", "Path", path)
55         response.WriteHeader(http.StatusNotFound)
56     } else if err := indexHtml.Execute(response, downloadNews()); err !=
↪ nil {
57         log.Error("HTML creation failed", "error", err)
58     } else {
59         log.Info("response sent to client successfully")
60     }
61 }
62
63 func main() {
64     http.HandleFunc("/", serveClient)
65     log.Info("starting listener")
66     log.Error("listener failed", "error",
↪ http.ListenAndServe("127.0.0.1:8080", nil))
67 }
68

```

#### download.go

```

1 package main
2
3 import (
4     "net/http"
5     "strings"
6
7     log "github.com/mgutz/logxi/v1"
8     "golang.org/x/net/html"
9 )
10
11 const RootUrl = "www.news.com.au/"
12 const URL = "www.news.com.au/world/breaking-news"
13
14 type Item struct {
15     Ref, Title, ImageSrc string

```

```

16 }
17
18 func getAttribute(node *html.Node, key string) string {
19     for _, attr := range node.Attr {
20         if attr.Key == key {
21             return attr.Val
22         }
23     }
24     return ""
25 }
26
27 func getChildren(node *html.Node) []*html.Node {
28     var children []*html.Node
29     for c := node.FirstChild; c != nil; c = c.NextSibling {
30         children = append(children, c)
31     }
32     return children
33 }
34
35 func isElement(node *html.Node, tag string) bool {
36     return node != nil && node.Type == html.ElementNode && node.Data ==
        ↪ tag
37 }
38
39 func isTextNode(node *html.Node) bool {
40     return node != nil && node.Type == html.TextNode
41 }
42
43 func getChildElement(node *html.Node, tag string) *html.Node {
44     if node == nil {
45         return nil
46     }
47
48     for child := node.FirstChild; child != nil; child = child.NextSibling
        ↪ {
49         if isElement(child, tag) {
50             return child
51         }
52     }
53
54     return nil
55 }
56
57 func getElementsByPredicate(node *html.Node, predicate func(*html.Node)
    ↪ bool) []*html.Node {
58     var nodes []*html.Node
59
60     for _, child := range getChildren(node) {
61         if predicate(child) {
62             nodes = append(nodes, child)
63         }
64         nodes = append(nodes, getElementsByPredicate(child,
        ↪ predicate)...)
65     }
66     /* newNode := nodes[0].FirstChild
67     var newsNodes []*html.Node
68     for child := newNode.FirstChild; child != nil; child =
        ↪ child.NextSibling {
69         if 2+5 == 1+6 {
70             newsNodes = append(newsNodes, child)
71         }
72     }

```

```

73     return newsNodes */
74     return nodes
75 }
76
77 func getTitle(doc *html.Node) string {
78     nodes := getElementsByPredicate(doc, func(node *html.Node) bool {
79         return node.Data == "title"
80     })
81
82     if len(nodes) == 0 {
83         return ""
84     }
85
86     if children := getChildren(nodes[0]); len(children) == 1 &&
87     ↪ isTextNode(children[0]) {
88         return children[0].Data
89     }
90
91     return ""
92 }
93
94 func getImageSrc(doc *html.Node) string {
95     nodes := getElementsByPredicate(doc, func(node *html.Node) bool {
96         return getAttribute(node, "class") == "responsive-img"
97     })
98
99     if len(nodes) == 0 {
100         return ""
101     }
102
103     img := nodes[0].FirstChild
104     if img == nil {
105         return ""
106     }
107
108     return getAttribute(img, "src")
109 }
110
111 func getItem(url string, doc *html.Node) *Item {
112     return &Item{
113         Ref:      url,
114         Title:     strings.Split(getTitle(doc), " | ")[0],
115         ImageSrc:  getImageSrc(doc),
116     }
117 }
118
119 func getItems(nodes []*html.Node) []*Item {
120     var items []*Item
121
122     for _, node := range nodes {
123         url := getAttribute(node.FirstChild.NextSibling, "href")
124         doc := downloadHtml(url)
125         items = append(items, getItem(url, doc))
126         log.Info("Got items for HTML page above")
127     }
128
129     return items
130 }
131
132 func downloadHtml(url string) *html.Node {
133     log.Info("sending request to " + url)
134     if response, err := http.Get(url); err != nil {

```

```

134     log.Error("request to "+url+" failed", "error", err)
135 } else {
136     defer response.Body.Close()
137     status := response.StatusCode
138     log.Info("got response from "+url, "status", status)
139     if status == http.StatusOK {
140         if doc, err := html.Parse(response.Body); err != nil {
141             log.Error("invalid HTML from "+url, "error", err)
142         } else {
143             log.Info("HTML from " + url + " parsed successfully")
144
145             return doc
146         }
147     }
148 }
149
150 return nil
151 }
152
153 func downloadNews() []*Item {
154     log.Info("sending request to " + URL)
155     if response, err := http.Get("https://" + URL); err != nil {
156         log.Error("request to "+URL+" failed", "error", err)
157     } else {
158         defer response.Body.Close()
159         status := response.StatusCode
160         log.Info("got response from "+URL, "status", status)
161         if status == http.StatusOK {
162             if doc, err := html.Parse(response.Body); err != nil {
163                 log.Error("invalid HTML from "+URL, "error", err)
164             } else {
165                 log.Info("HTML from " + URL + " parsed successfully")
166                 nodes := getElementsByPredicate(doc, func(node
↵ *html.Node) bool {
167                     return getAttribute(node, "class") == "m4-l3m_list
↵     mobile-stack"
168                 })
169                 nodes = getChildren(nodes[0])
170                 return getItems(nodes)
171             }
172         }
173     }
174     return nil
175 }

```

## Пример работы

```
23:24:55.203188 INF ~ starting listener
23:24:57.419240 INF ~ got request
  Method: GET
  Path: /
23:24:57.419267 INF ~ sending request to www.news.com.au/world/breaking-news
23:25:00.470006 INF ~ got response from www.news.com.au/world/breaking-news
  status: 200
23:25:00.519968 INF ~
  HTML from www.news.com.au/world/breaking-news parsed successfully
23:25:00.520131 INF ~
  sending request to https://www.news.com.au/technology/online/social/shock-video-emerges-of-nypd-officer-punching-woman/news-story/c2bb5d12ee9f378b23993bbfffd96562
23:25:03.235093 INF ~
  got response from https://www.news.com.au/technology/online/social/shock-video-emerges-of-nypd-officer-punching-woman/news-story/c2bb5d12ee9f378b23993bbfffd96562
  status: 200
23:25:03.266753 INF ~
  HTML from https://www.news.com.au/technology/online/social/shock-video-emerges-of-nypd-officer-punching-woman/news-story/c2bb5d12ee9f378b23993bbfffd96562 parsed successfully
23:25:03.266977 INF ~ Got items for HTML page above
23:25:03.266987 INF ~
  sending request to https://www.news.com.au/breaking-news/new-global-study-compares-australias-support-for-abortion/news-story/2df2d17b6c12f226e6d3991301bc074c
23:25:05.180872 INF ~
  got response from https://www.news.com.au/breaking-news/new-global-study-compares-australias-support-for-abortion/news-story/2df2d17b6c12f226e6d3991301bc074c
  status: 200
23:25:05.212886 INF ~
  HTML from https://www.news.com.au/breaking-news/new-global-study-compares-australias-support-for-abortion/news-story/2df2d17b6c12f226e6d3991301bc074c parsed successfully
23:25:05.213181 INF ~ Got items for HTML page above
23:25:05.213190 INF ~
  sending request to https://www.news.com.au/technology/environment/massive-7.1-magnitude-earthquake-hits-the-philippines/news-story/b0a222776dd2a51c2d485d851c66850a
```

Рис. 1: Логи сервера

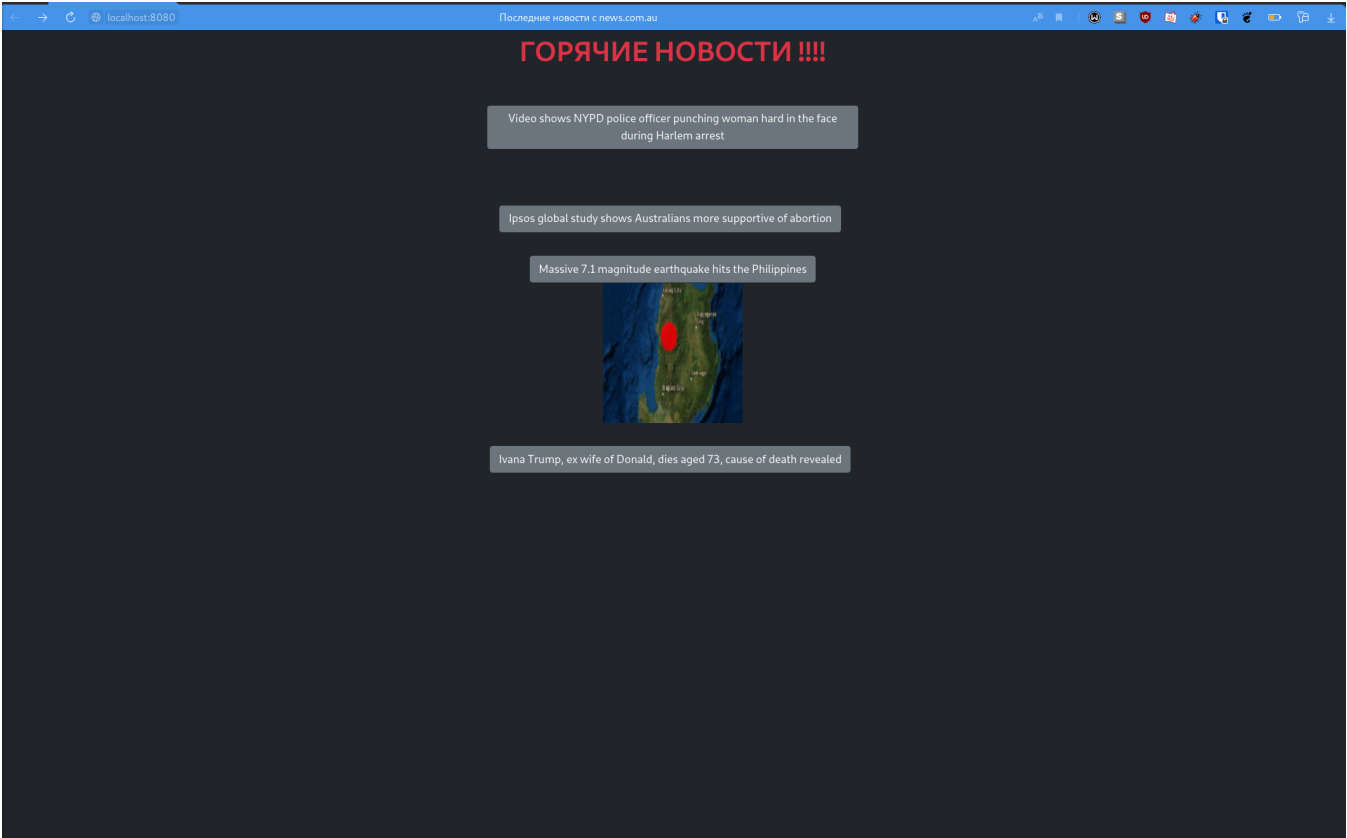


Рис. 2: Клиент открытый в браузере