|  |
| --- |
| package main |
|  |  |
|  | import ( |
|  | "fmt" |
|  | "log" |
|  | "net/http" |
|  |  |
|  | "golang.org/x/net/html" |
|  | ) |
|  |  |
|  | func main() { |
|  | http.HandleFunc("/", handler) |
|  | log.Fatal(http.ListenAndServe("0.0.0.0:8000", nil)) |
|  | } |
|  |  |
|  | func handler(w http.ResponseWriter, r \*http.Request) { |
|  | url := r.URL.Query().Get("q") |
|  | fmt.Fprintf(w, "Page = %q\n", url) |
|  | if len(url) == 0 { |
|  | return |
|  | } |
|  | page, err := parse("https://" + url) |
|  | if err != nil { |
|  | fmt.Printf("Error getting page %s %s\n", url, err) |
|  | return |
|  | } |
|  | links := pageLinks(nil, page) |
|  | for \_, link := range links { |
|  | fmt.Fprintf(w, "Link = %q\n", link) |
|  | } |
|  | } |
|  |  |
|  | func parse(url string) (\*html.Node, error) { |
|  | fmt.Println(url) |
|  | r, err := http.Get(url) |
|  | if err != nil { |
|  | return nil, fmt.Errorf("Cannot get page") |
|  | } |
|  | b, err := html.Parse(r.Body) |
|  | if err != nil { |
|  | return nil, fmt.Errorf("Cannot parse page") |
|  | } |
|  | return b, err |
|  | } |
|  |  |
|  | func pageLinks(links []string, n \*html.Node) []string { |
|  | if n.Type == html.ElementNode && n.Data == "a" { |
|  | for \_, a := range n.Attr { |
|  | if a.Key == "href" { |
|  | links = append(links, a.Val) |
|  | } |
|  | } |
|  | } |
|  | for c := n.FirstChild; c != nil; c = c.NextSibling { |
|  | links = pageLinks(links, c) |
|  | } |
|  | return links |
|  | } |