Working with HTTP Requests

Michael Van Sickle

@vansimke

Introduction

Query parameters

Form data

Working with JSON

Built-in responses

Query Parameters

https://localhost:8000/search?q=products&page=1

Query Parameters

https://localhost:8000/search?q=products&page=1

```
func (w http.ResponseWriter, r *http.Request) {
      url := r.URL                  // net/url.URL

      query := url.Query()          // Values (map[string][]string)

      q := query[“q”]               // []string{“products”}

      page := query.Get(“page”)     // “1”
}
```

Form Data

```
<form action=“.” method=“post”>
   Username: <input type=“text” name=“username”/>
   Password: <input type=“password” name=“password”/>
   <button type=“submit”>Submit</button>
</form>
```

Form Data

```
func (w http.ResponseWriter, r *http.Request) {
   err := r.ParseForm()      // populate Form and PostForm

   f := r.Form               // net/url.Values

   username := f[“username”] // []string {“mike”}

   pass := f.Get(“password”) // “password1”
}
```

Parsing Form Data

ParseForm

ParseMultipartForm

Reading Form Values

- Form
- PostForm
- FormValue()
- PostFormValue()
- FormFile()
- MultipartReader()

Working with JSON

```
{
   “term”: “products”,
   “page”: 1,
   “pageSize”: 25
}
```

```
type Query struct {
   Term     string
   Page     int
   PageSize int
}
```

Working with JSON

```
{
    “term”: “products”,
    “page”: 1,
    “pageSize”: 25
}
```

```
type Query struct {
    Term     string `json:”term”`
    Page     int    `json:”page”`
    PageSize int    `json:”pageSize”`
}
```

Reading JSON

```
type Query struct {
    Term     string  `json:”term”`
    Page     int     `json:”page”`
    PageSize int     `json:”pageSize”`
}
func (w http.ResponseWriter, r *http.Request) {
    dec := json.NewDecoder(r.Body)
    var query Query
    err := dec.Decode(&query)
}
```

Writing JSON

```
type Result struct {
    ID          int     `json:”id”`
    Name        string  `json:”name”`
    Description string  `json:”description”`
}
func (w http.ResponseWriter, r *http.Request) {
    var results []Result = model.GetResults()
    enc := json.NewEncoder(w)
    err := enc.Encode(results)
}
```

Built-in Responses

NotFound

ServeFile

ServeContent

Redirect

Summary

Query parameters

Form data

Working with JSON

Built-in responses