Middleware (Basic)

Middleware (Basic)

This example will show how to create basic logging middleware in Go.
A middleware simply takes a `http.HandlerFunc` as one of its parameters, wraps it and returns a new `http.HandlerFunc` for the server to call.

```go
// basic-middleware.go
package main

import (
    "fmt"
    "log"
    "net/http"
)

func logging(f http.HandlerFunc) http.HandlerFunc {
    return func(w http.ResponseWriter, r *http.Request) {
        log.Println(r.URL.Path)
        f(w, r)
    }
}

func foo(w http.ResponseWriter, r *http.Request) {
    fmt.Fprintln(w, "foo")
}

func bar(w http.ResponseWriter, r *http.Request) {
    fmt.Fprintln(w, "bar")
}

func main() {
    http.HandleFunc("/foo", logging(foo))
    http.HandleFunc("/bar", logging(bar))

    http.ListenAndServe(":8080", nil)
}
```

```
$ go run basic-middleware.go
2017/02/10 23:59:34 /foo
2017/02/10 23:59:35 /bar
2017/02/10 23:59:36 /foo?bar

$ curl -s http://localhost:8080/foo
$ curl -s http://localhost:8080/bar
$ curl -s http://localhost:8080/foo?bar
```

Legal Disclosure   Privacy Statement