

GOLOG

golog is a lightweight package that allows logging of requests and errors into a file
The http Logger takes in the filename permission and a handler
initialize a logger and call start

HTTP request logging

Log all your requests

NOTE This has not been tested on https

```
package main

import (
    "fmt"
    "log"
    "net/http"
    "github.com/3l0racle/golog"
)

func main(){
    ReqLog := golog.RequestLogger{
        FileName:"requests",
        Dir:"./data/logs/",
        Perm:0666,
        Handle:http.DefaultServeMux,
    }
    http.HandleFunc("/",RootHandler)
    err := http.ListenAndServe(fmt.Sprintf(":%d", 3000),golog.Start(ReqLog))
    if err != nil{
        log.Fatal(err)
    }
}

func RootHandler(res http.ResponseWriter, req *http.Request){
    fmt.Fprintf(res, "<h1>Hello World</h1><br><div>Welcome to wherever you are</div>")
}
```

Error Logging mechanisms

Later on will update log.Fatal and panic plus panic recovery

Log error to file in this specified directory

```
package main

import (
    "fmt"
    "errors"
    "github.com/3l0racle/golog"
)

func main(){
    err := errors.New("Test error logger final")
    // LogErrorToFileInDir("NameOfTheErrorFile","Directory","Error or fmt.Sprintf(Error of randomparameter %s %s
    golog.LogErrorToFileInDir("test","./data/logs/",err)
    randParameter := "This is that random thing I was saying"
    golog.LogErrorToFileInDir("test","./data/logs/",fmt.Sprintf("Error of %s plus error %s",randParameter,err))
}
```

```
//log error to file in the current directory
golog.LogErrorToFile("test",0666,errors.New("this is an error"))
}
```

Log error universally

PS :) I honestly don know why i created this it flipped my mind

```
package main

import (
    "errors"
    "github.com/3l0racle/golog"
)

func main(){
    err := errors.New("Test error logger")
    logggerr := golog.ErrorLogger{
        Filename: "test",
        Dir: "../data/logs/",
        Perm:0666,
        Text:err,
    }
    logggerr.UniversalLog()
}
```