

Response

```
func (r *Response) Buffer() []byte
func (r *Response) BufferLength() int
func (r *Response) BufferString() string
func (r *Response) Flush()
func (r *Response) SetBuffer(data []byte)
```

OutputFlush

```
package main

import (
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/net/ghttp"
    "net/http"
)

func MiddlewareErrorHandler(r *ghttp.Request) {
    r.Middleware.Next()
    if r.Response.Status >= http.StatusInternalServerError {
        r.Response.ClearBuffer()
        r.Response.Writeln("")
    }
}

func main() {
    s := g.Server()
    s.Group("/api.v2", func(group *ghttp.RouterGroup) {
        group.Middleware(MiddlewareErrorHandler)
        group.ALL("/user/list", func(r *ghttp.Request) {
            panic("db error: sql is xxxxxxxx")
        })
    })
    s.SetPort(8199)
    s.Run()
}
```

<http://127.0.0.1:8199/api.v2/user/list>