

gcode.Code

Content Menu

•
•

gcode.code

```
type BizCode struct {
    code    int
    message string
    detail  BizCodeDetail
}
type BizCodeDetail struct {
    Code    string
    HttpStatusCode int
}

func (c BizCode) BizDetail() BizCodeDetail {
    return c.detail
}

func (c BizCode) Code() int {
    return c.code
}

func (c BizCode) Message() string {
    return c.message
}

func (c BizCode) Detail() interface{} {
    return c.detail
}

func New(httpCode int, code string, message string) gcode.Code {
    return BizCode{
        code:    0,
        message: message,
        detail: BizCodeDetail{
            Code:    code,
            HttpStatusCode: httpCode,
        },
    }
}
```

```
var (
    CodeNil      = New(200, "OK", "")
    CodeNotFound = New(404, "Not Found", "Resource does not exist")
    CodeInternal = New(500, "Internal Error", "An error occurred
internally")
    // ...
)
```

```

func ResponseHandler(r *ghttp.Request) {
    r.Middleware.Next()
    // There's custom buffer content, it then exits current handler.
    if r.Response.BufferLength() > 0 {
        return
    }
    var (
        err  = r.GetError()
        res  = r.GetHandlerResponse()
        code = gerror.Code(err)
    )
    if code == gcode.CodeNil && err != nil {
        code = CodeInternal
    } else {
        code = CodeNil
    }
    if bizCode, ok := code.(BizCode); ok {
        r.Response.WriteStatus(bizCode.BizDetail().HttpCode)
    }
    r.Response.WriteJson(g.Map{
        `code`:    gcode.CodeOK.Code(),
        `message`: gcode.CodeOK.Message(),
        `data`:    res,
    })
}

```