

## Content Menu

- [,API.](#)
- [:codmsg](#)

,API.

API apiv1/hello.go

: reqres API! ,

LoginDoReq LoginDoRes handler!

```
type LoginDoReq struct {
    g.Meta `path:"/login" method:"post" summary:"" tags:"" `
    Name    string `json:"name" in:"query" d:"admin" dc:"" `
    Password string `json:"password" in:"query" d:"123456" dc:"()" `
}
type LoginDoRes struct {
    g.Meta `mime:"json" example:"string" `
    Token string `json:"token" dc:"token" `
}
```

internal/handler/hello.go

: return,

```
func (h *hHello) Login(ctx context.Context, req *apiv1.LoginDoReq)(res
*apiv1.LoginDoRes, err error) {

    token := req.Name + req.Password
    g.Dump(token)

    res = &apiv1.LoginDoRes{
        Token: token,
    }
    return
}
```

internal/cmd/cmd.go Login,-

:()

”

```
g.RequestFromCtx(ctx).GetMap()
g.RequestFromCtx(ctx).GetHeader()
```

```
g.RequestFromCtx(ctx).Response.WriteJson(g.Map{"token": "token"})
```

:codmsg

ghttp.MiddlewareHandlerResponse !

```
type DefaultHandlerResponse struct {
    Code    int        `json:"code"`
    Message string      `json:"message"`
    Data    interface{} `json:"data"`
}

//
func MiddlewareGuRes(r *ghttp.Request) {
    r.Middleware.Next()
    // ,
    //https://goframe.org/pages/viewpage.action?pageId=1114281
    if r.Response.BufferLength() > 0 {
        return
    }
    //
    var (
        err    error
        res    interface{}
    )
    //
    res, err = r.GetHandlerResponse()
    if err != nil {
        code := gerror.Code(err) //
        if code == gcode.CodeNil {
            code = gcode.CodeInternalServerError
        }
        r.Response.WriteJson(DefaultHandlerResponse{
            Code:    CodeInternalServerError.code, // 500
            Message: fmt.Sprintf(CodeInternalServerError.msg, "
: ",err.Error()," ",code.Message()),
            Data:    nil,
        })
        return
    }
    //
    r.Response.WriteJson(DefaultHandlerResponse{
        Code:    CodeOK.code, // 200
        Message: CodeOK.msg,
        Data:    res,
    })
}
```