-

Detail

Content Menu

-
  -
  -
-


```
type BizCode struct {
        User User
        // ...
}

type User struct {
        Id   int
        Name string
        // ...
}
```

WithCode

```
// WithCode creates and returns a new error code based on given Code.
// The code and message is from given `code`, but the detail if from given
`detail`.
func WithCode(code Code, detail interface{}) Code
```

```
code := gcode.WithCode(gcode.CodeNotFound, BizCode{
        User: User{
                Id:   1,
                Name: "John",
        },
})
fmt.Println(code)
```

code

```
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 = gcode.CodeInternalError
        }
        if detail, ok := code.Detail().(BizCode); ok {
                g.Log().Errorf(r.Context(), `error caused by user "%+v"`,
detail.User)
        }
        r.Response.WriteJson(ghttp.DefaultHandlerResponse{
                Code:    gcode.CodeOK.Code(),
                Message: gcode.CodeOK.Message(),
                Data:    res,
        })
}
```



ServerDetail