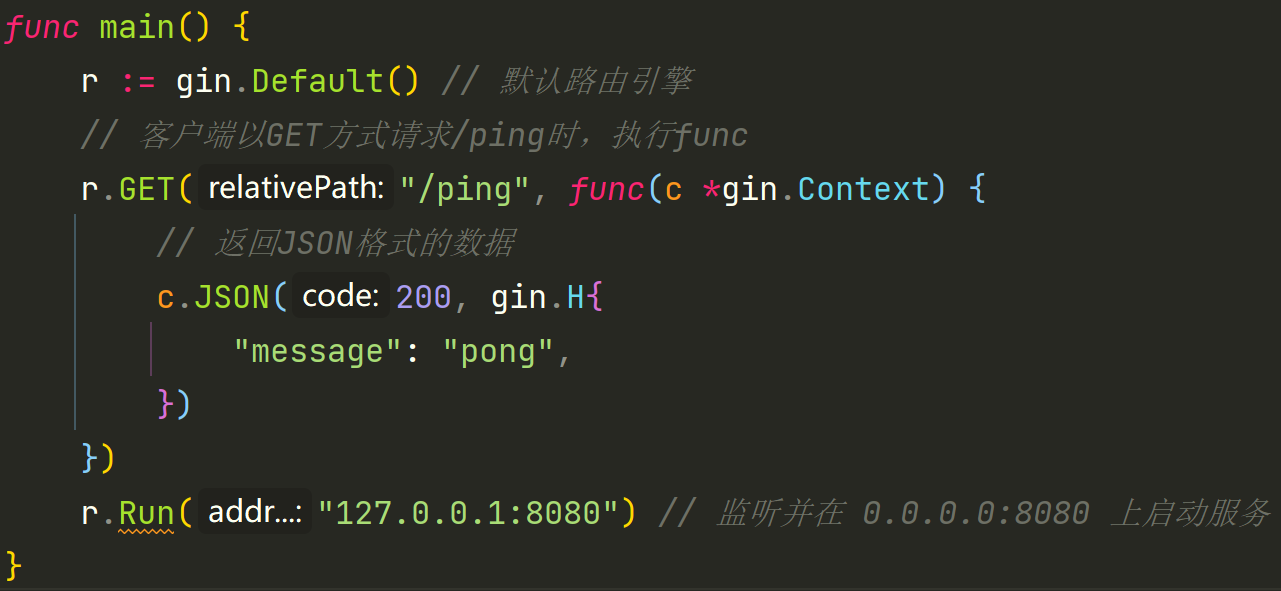
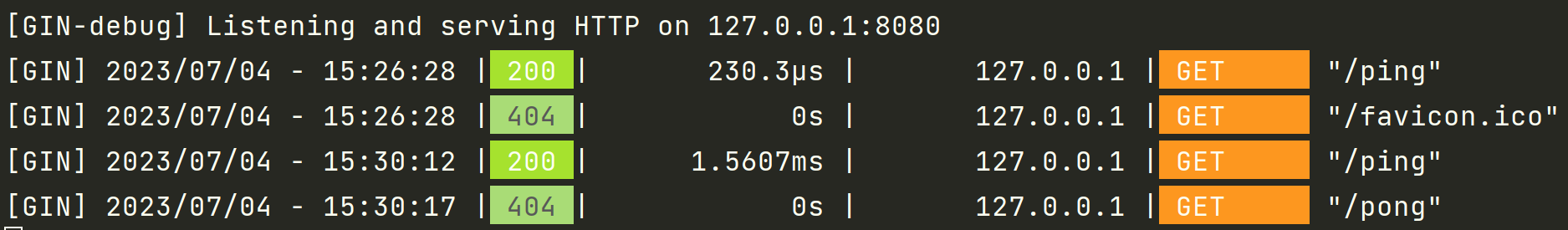
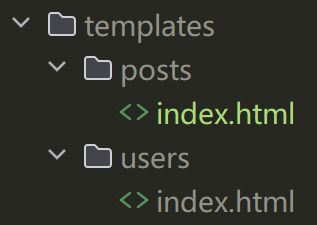
# Gin

1. Gin是一个Golang中使用的开源的Web框架，可以方便的创建web服务器，制定服务器的业务逻辑
2. 链接
   1. Gin框架介绍及使用：[Gin框架介绍及使用 | 李文周的博客 (liwenzhou.com)](https://www.liwenzhou.com/posts/Go/Gin_framework/#autoid-0-6-0)
   2. RESTFul API：​[RESTFul API](https://mubu.com/doc2RanSh9U7E0)​
3. 使用方法
   1. go get -u [github.com/gin-gonic/gin](http://github.com/gin-gonic/gin)
   2. 第一个Gin示例：





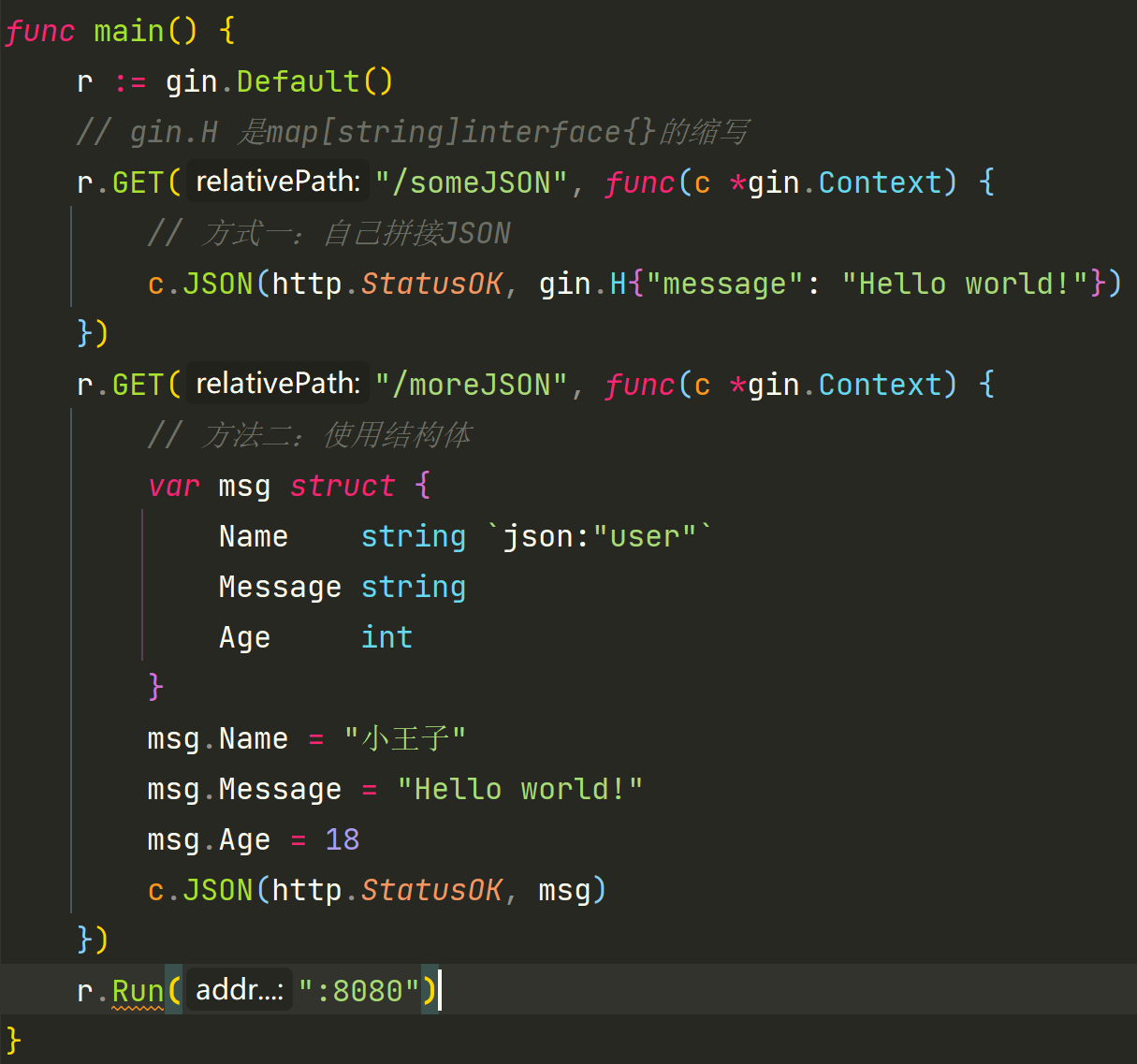
* 1. 渲染
     1. 模板：例如一些html文件，用于渲染网页



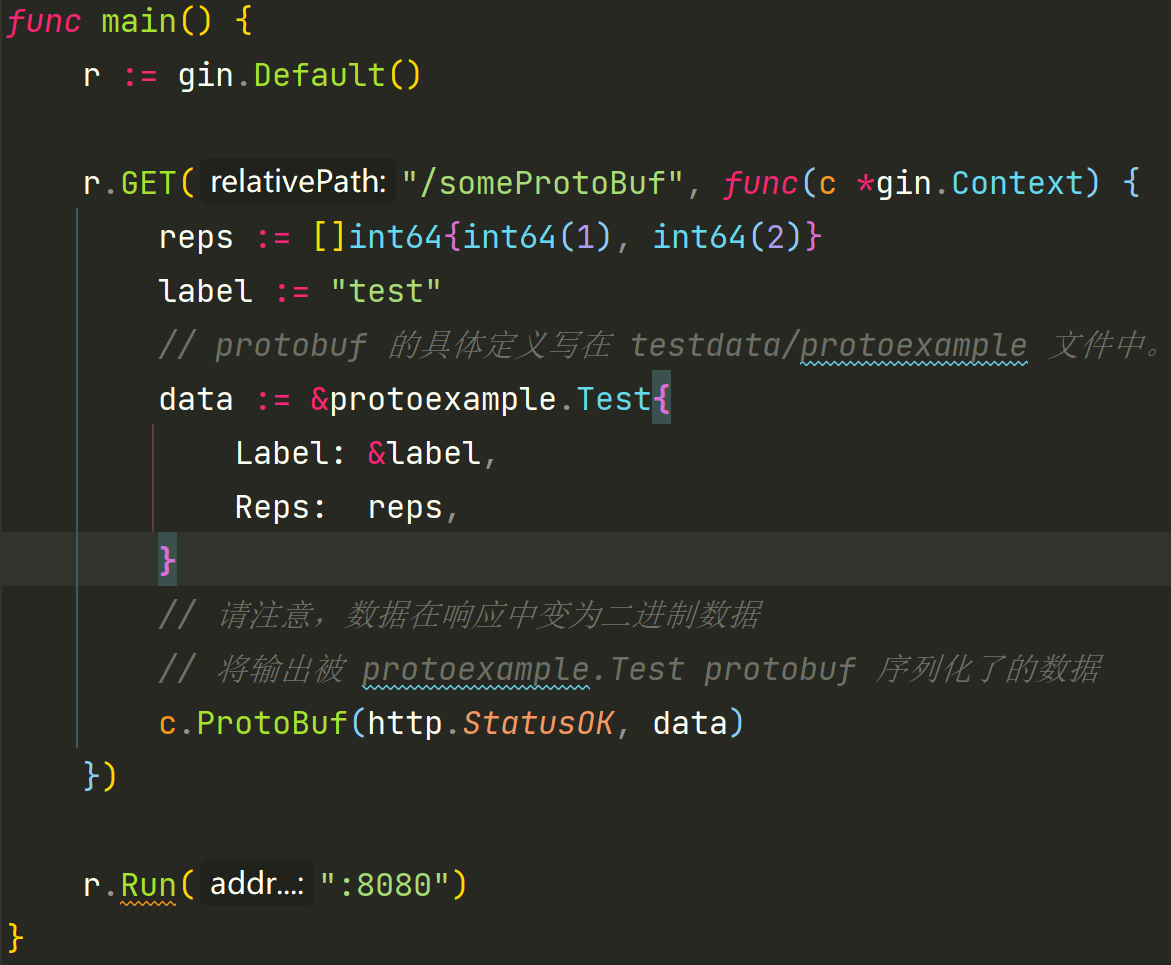
* + 1. HTML渲染



* + 1. JSON渲染



* + 1. protobuf渲染



* 1. URL参数：query，form，json，path
     1. query
        1. querystring指的是URL中?后面携带的参数，例如：/user/search?username=小王子&address=沙河。 获取请求的querystring参数的方法如下：



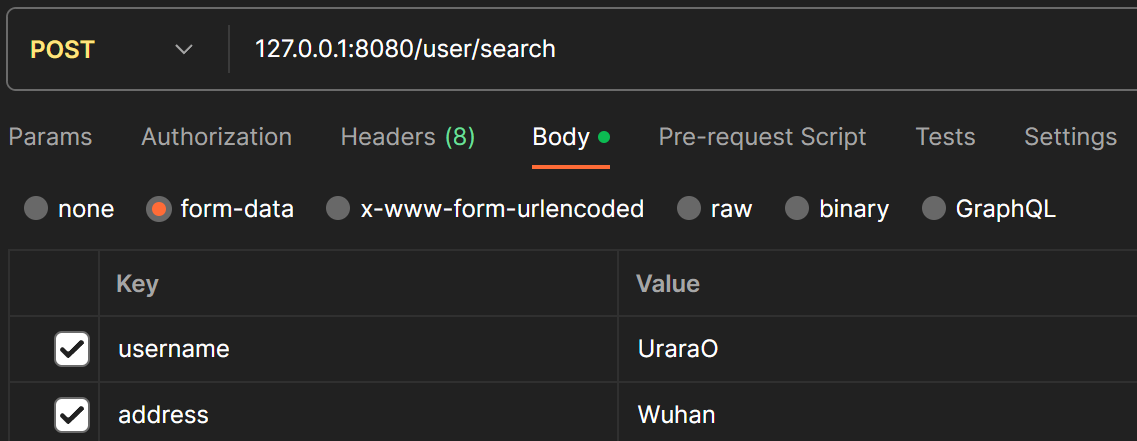
* + - * 1. 请求如下



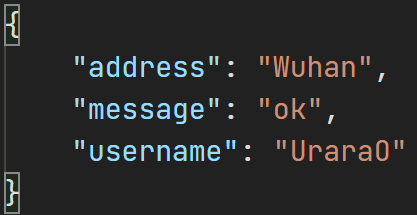
* + - * 1. 服务端得到的username = 小王子，address = 沙河，即从客户端发送的URL中提取出询问参数
    1. form
       1. 前端通过POST方法提交表单，则服务端需要提取表单数据进行后续操作：



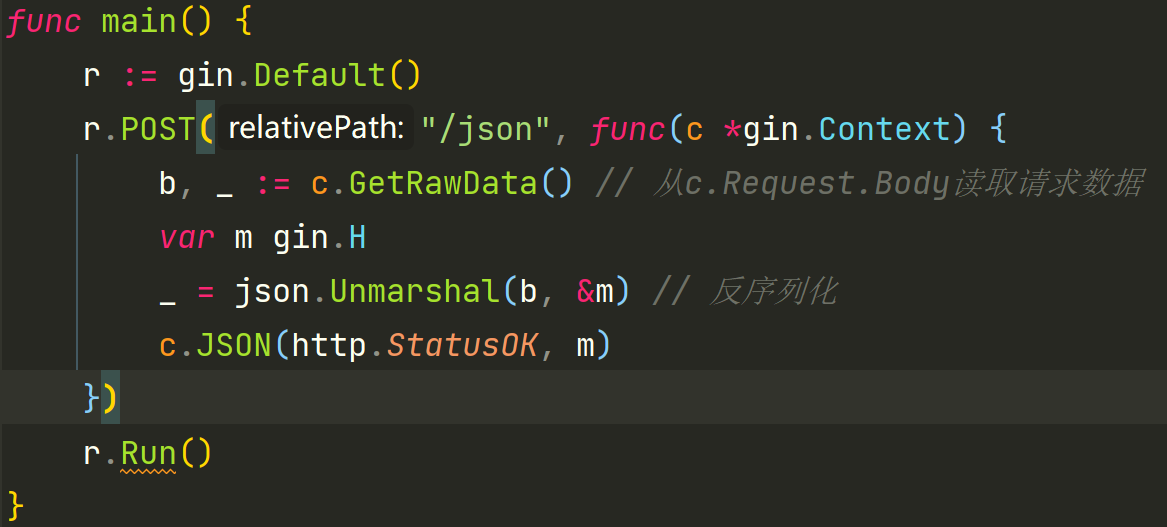
* + - * 1. 前端发送表单方式如下



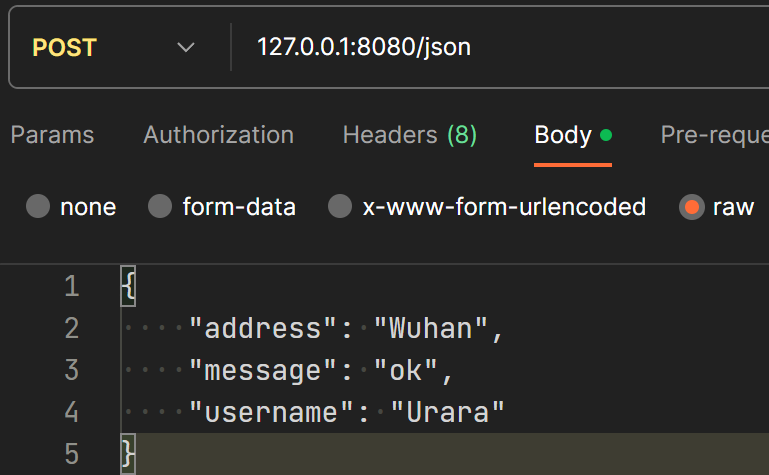
* + - * 1. 结果



* + 1. json
       1. 前端请求的数据通过json提交，则服务端处理如下：



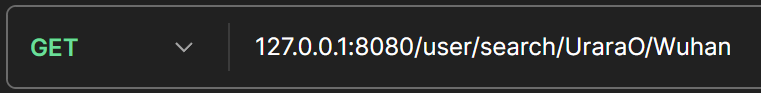
* + - 1. 前端请求：



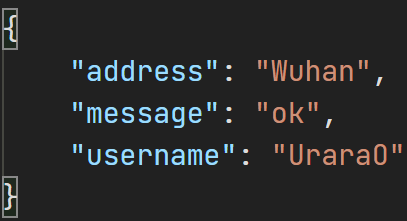
* + - 1. 结果与发送数据相同，将json报文原样提取，并返回
    1. path
       1. 请求的path通过URL路径传递，例如：/user/search/小王子/沙河。 获取请求URL路径中的参数的方式如下：



* + - 1. 请求



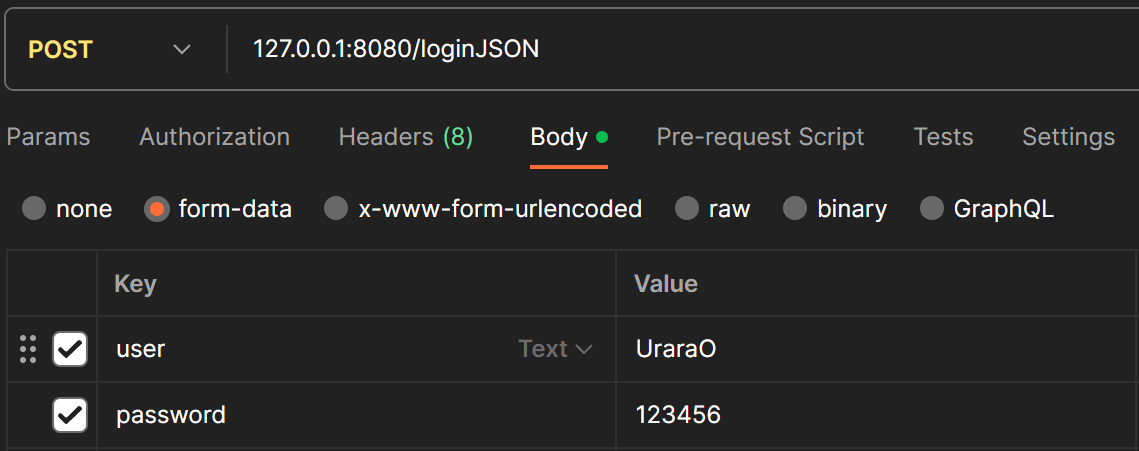
* + - 1. 结果



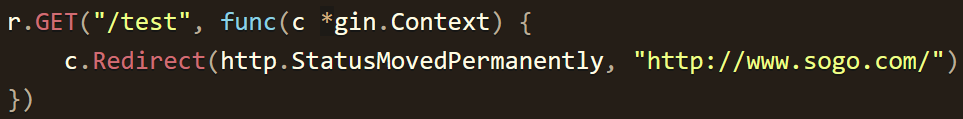
* + 1. 参数绑定
       1. 参数绑定利用反射机制，自动将报文中的数据绑定到结构体中，方便使用
       2. 服务端用法：定义反射数据结构体，用于接收数据



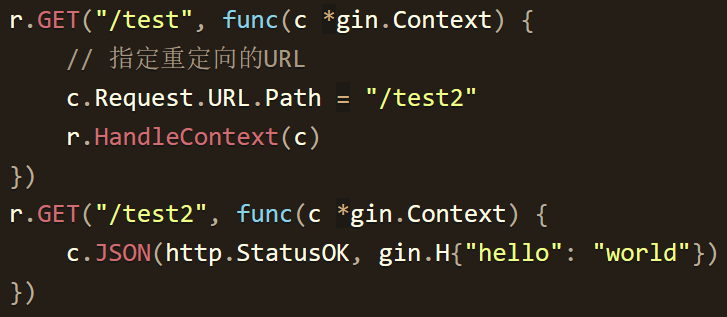
* + - 1. 前端请求：



* 1. 重定向
     1. http重定向



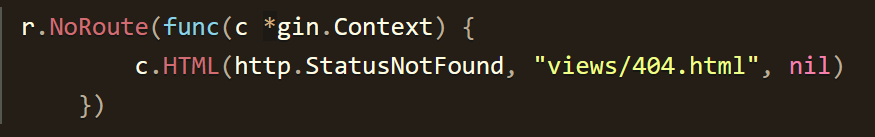
* + 1. 路由重定向



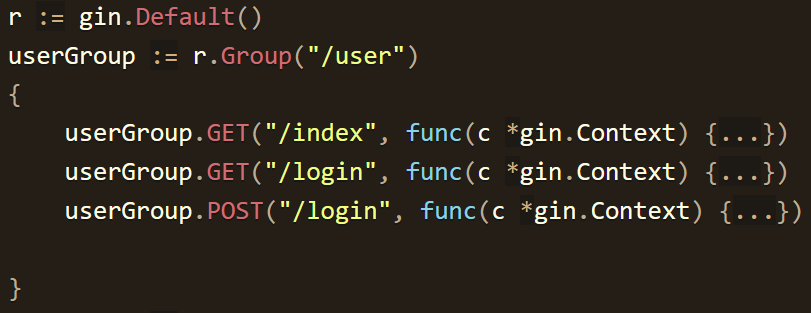
* 1. Gin路由
     1. GET，POST等方法如前所示，还有可以匹配任意方法的Any



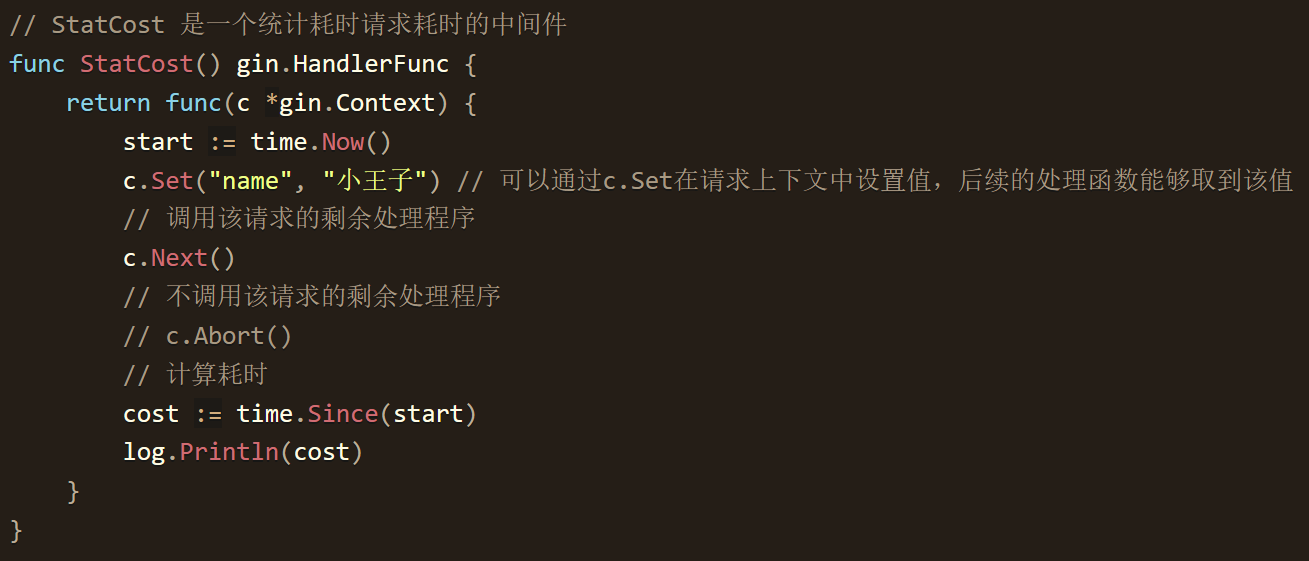
* + 1. 若不匹配任何方法，404 Not Found



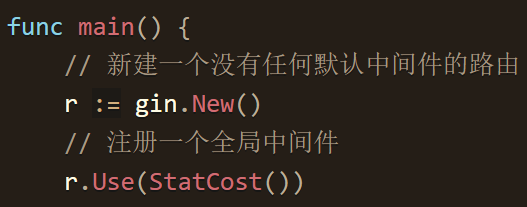
* + 1. 路由组



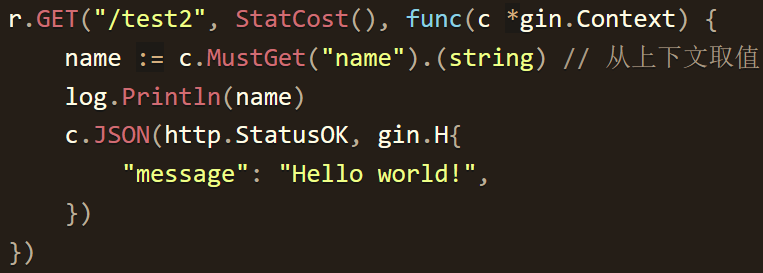
* 1. Gin中间件
     1. 定义中间件：中间件必须返回 gin.HandlerFunc 类型，即返回一个可调用对象func(\*gin.Context)



* + 1. 注册中间件
       1. 全局路由



* + - 1. 某个路由



* + - 1. 路由组

