

Please visit website: <http://cxyroad.com>

byteimg.com/tos-cn-i-k3u1fbpfcp/e851452e1a7343849e3239c3aac0ab50~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2560&h=1774&s=335861&e=png&b=f7f6f9)

这是创建的web服务代码目录，已经包含了teacher表的CRUD api所有代码，包含了Gin和Gorm的初始化和配置代码，开箱即用。

...

```
.
├── cmd
│   └── user
│       ├── initial
│       └── main.go
├── configs
├── deployments
│   ├── binary
│   ├── docker-compose
│   └── kubernetes
├── docs
├── internal
│   ├── cache
│   ├── config
│   ├── dao
│   ├── ecode
│   ├── handler
│   ├── model
│   ├── routers
│   ├── server
│   └── types
└── scripts
```

...

解压代码文件，打开终端，切换到web服务代码目录，执行命令：

...

```
# 生成swagger文档
make docs
```

```
# 编译和运行服务
make run
```

```
...
```

在浏览器打开

[<http://localhost:8080/swagger/index.html>](<http://cxyroad.com/>  
"http://localhost:8080/swagger/index.html"), 可以在页面上进行增删改查  
api测试, 如下图所示:

![web-http-swagger.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/07eb2c40d5404a988c100d048d15b4b6~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2560&h=2427&s=163140&e=png&b=faf8f8)

### ### 批量添加CRUD api

如果有新的mysql表需要生成CRUD api代码,

1. 点击左边菜单栏【Public】-->【生成handler CRUD代码】;
2. 选择数据库`mysql`, 填写`数据库dsn`, 然后点击按钮`获取表名`, 选择mysql表(可多选);
3. 填写其他参数。

填写完参 密码

```
// 生成和存储token
```

```
response.Success(c, gin.H{
    "id": 1,
    "token": "xxxxxx",
})
}
```

```
...
```

然后把Login方法添加到TeacherHandler接口:

```
...
type TeacherHandler interface {
Create(c *gin.Context)
// ...
Login(c *gin.Context)
}
...
```

#### **\*\*4. 注册路由\*\***

进入目录`internal/routers`，打开文件`teacher.go`，把Login路由注册进来：

```
...
func teacherRouter(group *gin.RouterGroup, h handler.TeacherHandler) {
group.POST("/teacher", h.Create)
// ...
group.POST("/teacher/login", h.Login)
}
...
```

#### **\*\*5. 编写业务逻辑代码\*\***

编写登录的业务逻辑代码，例如验证密码、生成token等。

> 在人工添加的自定义api中，可能需要对数据增删改查操作(也叫dao CRUD)，这些dao CRUD代码可以直接生成，不需要人工编写，直接在在sponge的UI界面生成： **\*\*Public\*\* --> \*\*生成dao CRUD代码\*\***

#### **\*\*6. 测试api\*\***

编写完业务逻辑代码后，在终端执行命令：

...

```
# 生成swagger文档  
make docs
```

```
# 编译和运行服务  
make run
```

...

在浏览器刷新

[<http://localhost:8080/swagger/index.html>](<http://cxyroad.com/>  
"http://localhost:8080/swagger/index.html"), 在页面上可以看到登录  
api, 在页面测试通过后, 一个自定义的api才算正式添加完成。

### ### 总结

sponge是集成了Gin 和 Gorm 强大的web后端服务开发工具, 可以帮助开发者轻松快速构建 RESTful API 服务。希望能够帮助 Golang 开发者彻底告别手写 CRUD api, 更轻松便捷的构建高效、优雅的 RESTful API 服务。

原文链接: <https://juejin.cn/post/7360892717545291785>