

# 1, 流程

1. 基于大明老师第6课的Live分支改写的作业，其中将/users/edit的Endpoint改成了/users/profile/edit。因为觉得这个是对profile的编辑。
2. 新建了一个user\_profile的表来保存user profile信息，相应的增加了UserProfileHandler等一套service, repo, dao...
3. 对前端不熟悉，所以完全用Apifox (类Postman工具)测试。

在做作业的过程中有以下几个疑问，希望老师或者助教老师能解惑下：

1, 我的userid是从session里拿到的，我的理解是这个userid是用户登陆成功后将user\_id保留在session里的，而这个session是保留在Gin的context里的，所以一旦程序关闭，这个session就不存在了。

但是我的测试结果是：我重启了程序后，还是能拿到我上次退出程序后登录的user id，这是为什么？

2, 关于session我的理解是当浏览器登录到服务器时，会带上cookie里的session id，服务端会校验这个session id找到服务器里保存的session，然后得到保存的一些信息，比如user id之类。但是我在程序里没有看到校验session id的步骤，在检查登录态时直接是通过语句：sess := sessions.Default(ctx)来拿到session的，那这不是只要服务端有session都能通过检查？那怎么保证拿到的user id确实是用户的user id呢？

3, 错误码的管理太过复杂了，基本上增加一个错误码就会在dao, repo, service层都要加一个，这个有什么好的解决方案吗？

## 2, Signup

输入用户信息，提示“注册成功”

POST http://localhost:8077/users/signup

Params Body 1 Headers 1 Cookies Auth 前置操作 后置操作 设置

none form-data x-www-form-urlencoded json xml raw binary GraphQL msgpack

动态值 提取 格式化

```
1 {
2   "email": "test1@qq.com",
3   "password": "a12345678@",
4   "confirmPassword": "a12345678@"
5 }
```

Body Cookie Header 3 控制台 实际请求

Pretty Raw Preview Visualize Text utf8

1 注册成功

校验响应 成功 (200)

200 473 ms 12 B

查看Mysql，该记录已经写入数据库，密码已加密。

属性 数据 ER 图 webook 数据库					
Users 输入一个SQL表达式来过滤结果(使用Ctrl+Space)					
id	email	password	ctime	utime	
1	test1@qq.com	\$2a\$10\$GTnWdmHxMcBNPajMwqxMA.PWGy	1,691,740,600,566	1,691,740,600,566	

## 3, Login

POST http://localhost:8077/users/login
发送
暂存
保存为用例

Params
Body 1
Headers 1
Cookies
Auth
前置操作
后置操作
设置

none form-data x-www-form-urlencoded json xml raw binary GraphQL msgpack

自动生成
动态值
提取
格式化

```

1 {
2   "email": "test1@qq.com",
3   "password": "a12345678@"
4 }

```

Body
Cookie
Header 4
控制台
实际请求

Pretty
Raw
Preview
Visualize
Text
utf8
提取

1 登录成功

校验响应 成功 (200)
200 63 ms 12 B

## 4, Profile

### 4.1 测试一个不存在的Profile

这里profile的user id是从session储存的user\_id里拿的:

```

func getUserId(ctx *gin.Context) (any, error) {
    // 需要从session里拿到userId
    sess := sessions.Default(ctx)
    uid := sess.Get("userId")
    if uid == nil {
        // 没有登录
        ctx.AbortWithStatus(http.StatusUnauthorized)
        return nil, ErrNotLogin
    }
    return uid, nil
}

```

得到userid后再从user\_profile表里拿到UserProfile信息。这里如果没有找到UserProfile, 我默认没有错误, 因为没有user profile是一个很正常的情况, 所以直接将3个空属性返回上来。

GET http://localhost:8077/users/profile

发送暂存保存为用例

ParamsBodyHeadersCookies1Auth前置操作后置操作设置

Query 参数

参数名	参数值	类型	说明
添加参数			

BodyCookieHeader3控制台实际请求

PrettyRawPreviewVisualizeTextutf8提取

1 这是你的 Profile:

2 昵称是: , 生日是:, 简介是

校验响应成功 (200)

200 9 ms 58 B

返回数据结构与接口定义不一致

## 4.2 测试一个存在的Profile

GET http://localhost:8077/users/profile

ParamsBodyHeadersCookies1Auth前置操作后置操作设置

参数名	参数值	类型
webook	MTY5MTc0NzkzNXxEdi1CQkFFQ180SL	string
添加参数		

BodyCookieHeader3控制台实际请求

PrettyRawPreviewVisualizeTextutf8提取

1 这是你的 Profile:

2 昵称是: bestman, 生日是:2003-02-01, 简介是: This is my updated webook

## 5, Profile Edit

### 5.1, 正确的更新

三个属性都符合条件的更新：

POST http://localhost:8077/users/profile/edit 发送 暂存 保存为用例

Params Body 1 Headers 1 Cookies Auth 前置操作 后置操作 设置 不一致

none form-data x-www-form-urlencoded json xml raw binary GraphQL msgpack

```
{ 1 2   "nick_name": "bestman", 3   "birthday": "2003-02-01", 4   "intro": "This is my webook." 5 }
```

提取 格式化

Body Cookie Header 3 控制台 实际请求

校验响应 成功 (200)

200 10.65 s 24 B

1 更新用户信息成功

查询数据库user\_profile表:

user\_profiles 输入一个SQL表达式来过滤结果 (使用Ctrl+Space)

	id	user_id	nick_name	birthday	intro	ctime	utime
1	9	1	bestman	2003-02-01	This is my webook	1,691,759,413,432	1,691,759,413,432

## 5.2, 昵称过长

昵称长度不应该超过16。

POST http://localhost:8077/users/profile/edit 发送 暂存

Params Body 1 Headers 1 Cookies Auth 前置操作 后置操作 设置

none form-data x-www-form-urlencoded json xml raw binary GraphQL msgpack

```
{ 1 2   "nick_name": "&cDbBQ-Y2z2Gm1jUDw*9", 3   "birthday": "2003-02-01", 4   "intro": "This is my webook." 5 }
```

提取

Body Cookie Header 3 控制台 实际请求

校验响应 成功

200 3.84 s 18 B

1 昵称格式不对

## 5.3, 生日格式不对

生日的格式应为“YYYY-MM-DD”

POSThttp://localhost:8077/users/profile/edit

发送

暂存

ParamsBody 1Headers 1CookiesAuth前置操作后置操作设置

noneform-datax-www-form-urlencodedjsonxmlrawbinaryGraphQLmsgpack

\* 动态值

1{

2  "nick\_name": "bestman",

3  "birthday": "20030201",

4  "intro": "This is my webook."

5}

BodyCookieHeader 3控制台实际请求

校验响应 200 8 ms 18 B

PrettyRawPreviewVisualizeText utf8提取

1生日格式不对

## 5.4, 简介过长

简介长度不应该超过128

POSThttp://localhost:8077/users/profile/edit

发送

暂存

ParamsBody 1Headers 1CookiesAuth前置操作后置操作设置

noneform-datax-www-form-urlencodedjsonxmlrawbinaryGraphQLmsgpack

\* 动态值

2  "nick\_name": "bestman",

3  "birthday": "2003-02-01",

4  "intro": "z\_H+QC7=fq1XstD3&1\_\_37Bw1Cyz=78\_

5  "intro": "z\_H+QC7=fq1XstD3&1\_\_37Bw1Cyz=78\_

BodyCookieHeader 3控制台实际请求

校验响应 200 3 ms 18 B

PrettyRawPreviewVisualizeText utf8提取

1简介格式不对

## 5.5, 更新profile信息

在已有profile信息基础上更新profile

POST

http://localhost:8077/users/profile/edit

发送

Params

Body1

Headers1

Cookies

Auth

前置操作

后置操作

设置

none

form-data

x-www-form-urlencoded

json

xml

raw

binary

GraphQL

msgpack

\* 动态值

1

{

2

"nick\_name": "bestman",

3

"birthday": "2003-02-01",

4

"intro": "This is my updated webook"

5

}

Body

Cookie

Header3

控制台

实际请求

校验响应

Pretty

Raw

Preview

Visualize

Text

utf8

提取

下载

复制

搜索

1

更新用户信息成功

200

12 r

返回数

查看user\_profile表，简介更新成功

user_profiles 输入一个SQL表达式来过滤结果(使用Ctrl+Space)								
	id	user_id	nick_name	birthday	intro	ctime	utime	
1	9	1	bestman	2003-02-01	This is my updated webook	1,691,759,413,432	1,691,759,448,913	