

个人开发日志-孔欣然

12.17

1.设计项目内容，组内分工，设计数据库

12.19 完成用户注册与登录模块接口与前端基础

1.建立数据库campus_lost_found

2.建立表user，具体字段如下：

字段名	类型	说明
id	BIGINT	主键，自增
username	VARCHAR(50)	用户名
password	VARCHAR(255)	密码（加密存储）
contact	VARCHAR(100)	联系方式（微信号）
role	VARCHAR(20)	用户角色（user / admin）
create_time	DATETIME	注册时间

3.完成注册接口，并测试

The screenshot shows a POST request to the URL `http://localhost:8080/api/register`. The request body is a JSON object:

```
1 {  
2   "username": "test1",  
3   "password": "123456",  
4   "contact": "skjffabw"  
5 }  
6
```

The response status is `200 OK`, and the message is `注册成功`.

4.完成 JWT 登录接口，并测试

The screenshot shows the Postman interface with a successful API call. The top bar indicates a **POST** method and the URL **http://localhost:8080/api/login**. The **Body** tab is selected, showing the JSON input:

```
1 {  
2   "username": "test",  
3   "password": "123456"  
4 }  
5
```

The response section shows a **200 OK** status with a response time of 195 ms and a response size of 538 B. The response body is displayed as JSON:

```
1 {  
2   "role": "user",  
3   "token": "eyJhbGciOiJIUzI1NiJ9.  
        eyJzdWIiOiJ0ZXN0Iiwicm9sZSI6InVzZXIiLCJpYXQiOjE3NjYxMTQ2MjAsImV4  
        cCI6MTc2NjIwMTAyMH0.  
        mri-paLeDBdtgFvp0TxuYSe8NwOu0kRAH8JNNNL39hmM",  
4   "username": "test"  
5 }
```

5.完成登录状态保持和受保护接口

测试受保护接口：

The screenshot shows the Postman interface with the following details:

- Method: GET
- URL: http://localhost:8080/api/profile
- Headers tab selected, showing:
 - Authorization: Bearer <token>
- Body tab selected, showing a JSON response:

```
1 {  
2   "role": "user",  
3   "username": "test2"  
4 }
```
- Status: 200 OK
- Time: 310 ms
- Size: 455 B

6. 完整注册登录流程：

1. 用户注册: /api/register (密码加密)
2. 用户登录: /api/login (验证密码 + 返回 JWT)
3. 前端保存 token, 后续请求都带 Authorization 头
4. 后端通过 JwtFilter 验证 token, 受保护接口才能访问
5. 受保护接口从 request.getAttribute("username") 获取登录信息

7. 前端

完成注册页、登录页、简单profile界面（登录状态保持，显示用户名）

登录

用户名

请输入用户名

密码

请输入密码

登录

没有账号? [注册](#)

注册

用户名

请输入用户名

密码

请输入密码

联系方式

请输入联系方式

注册

已有账号？登录

用户信息

用户名： test3

角色： user

退出登录

12.22

1.建立表lost_item

2.完成后端接口并测试

发布失物 (POST /lost_item)

编辑失物 (PUT /lost_item/{id})

删除失物 (DELETE /lost_item/{id})

列表查询 (GET /lost_item)

详情查询 (GET /lost_item/{id})

3.完成失物列表前端

失物列表

标题	类别	丢失地点	状态	操作
丢失校园卡（更新）	证件	图书馆二楼	已找回	<button>查看</button>
丢失钥匙	钥匙	A楼	未找回	<button>查看</button>
丢失钱包	钱包	C楼	未找回	<button>查看</button>

4.完成首页页面

5.失物信息前端（详情页、添加、编辑）

丢失水杯

[返回失物列表](#)

发布者：test

类别：个人物品

地点：综合楼三楼

丢失时间：2025-12-24 02:21

描述：蓝色水杯，上面有水母图案

未找回

编辑

标记已找回

删除

编辑失物信息

* 标题	丢失水杯
类别	个人物品
丢失地点	综合楼三楼
丢失时间	(2025-12-24 02:21:00)
描述	蓝色水杯，上面有水母图案

保存修改

发布失物信息

* 标题	
类别	
丢失地点	
丢失时间	()
描述	

发布

6.解决时间字段的类型不匹配问题

后端实体 LostItem 中是

```
private LocalDateTime lostTime;
```

前端现在提交

```
lostTime: dayjs(form.value.lostTime).format('YYYY-MM-DD HH:mm:ss')
```

会导致Spring Boot 的Jackson **无法自动把这种字符串（没有时区）转换成 LocalDateTime**，返回 400

解决：

后端实体保持 LocalDateTime

前端处理 lostTime,手动格式化为 yyyy-MM-dd HH:mm 字符串再提交

12.26

1.完成个人中心前端（失物部分）

The screenshot shows a web application interface for a university lost and found system. At the top, there is a blue header bar with the text "校园失物招领平台" on the left, a user profile "newuser" in the middle, and a "返回首页" (Return to Home) button on the right. Below the header, there are two main navigation tabs: "我发布的失物" (Items I Published) and "我发布的招领" (Items I Found). The "我发布的失物" tab is currently active and displays a table of lost items. The table has columns for Title, Category, Lost Location, Lost Time, Status, and Operations. There are two entries:

标题	类别	丢失地点	丢失时间	状态	操作
丢失校园卡 (更新)	证件	图书馆四楼	2025-12-26 15:30	已找回	<button>编辑</button> <button>删除</button>
丢失水杯	个人物品	综合楼三楼	2025-12-24 11:33	未找回	<button>编辑</button> <button>删除</button>

2.调整页面逻辑

3.完成页面美化