

浙江大学

本科实验报告

课程名称: B/S 体系软件设计

姓 名: 陈忆杭

学 院: 计算机科学与技术学院

专 业: 计算机科学与技术

学 号: 3210103981

指导教师: 胡晓军老师

2024 年 1 月 4 日

浙江大学实验报告

课程名称： B/S 体系软件设计 实验类型： 综合

实验项目名称： 物联网设备管理平台

学生姓名： 陈忆杭 专业： 计算机科学与技术 学号： 3210103981

指导老师： 胡晓军老师 实验日期： 2024 年 1 月 4 日

1. 概述

测试报告的测试对象是在《B/S 体系软件设计》课程中完成的课程项目《物联网设备管理平台》，本测试报告记录的是对该软件功能性测试的过程和结果。测试的功能包括：

- 注册新用户
- 已有用户登录
- 查看并修改用户注册时提交的个人信息和密码
- 创建和修改设备配置
- 查看和搜索用户创建过的设备
- 在地图上显示过去一段时间某设备的运行状况

在之前的开发过程中，对后端 API 已经使用 Postman 做过测试，因此这里我们直接用浏览器测试。

2. 测试环境

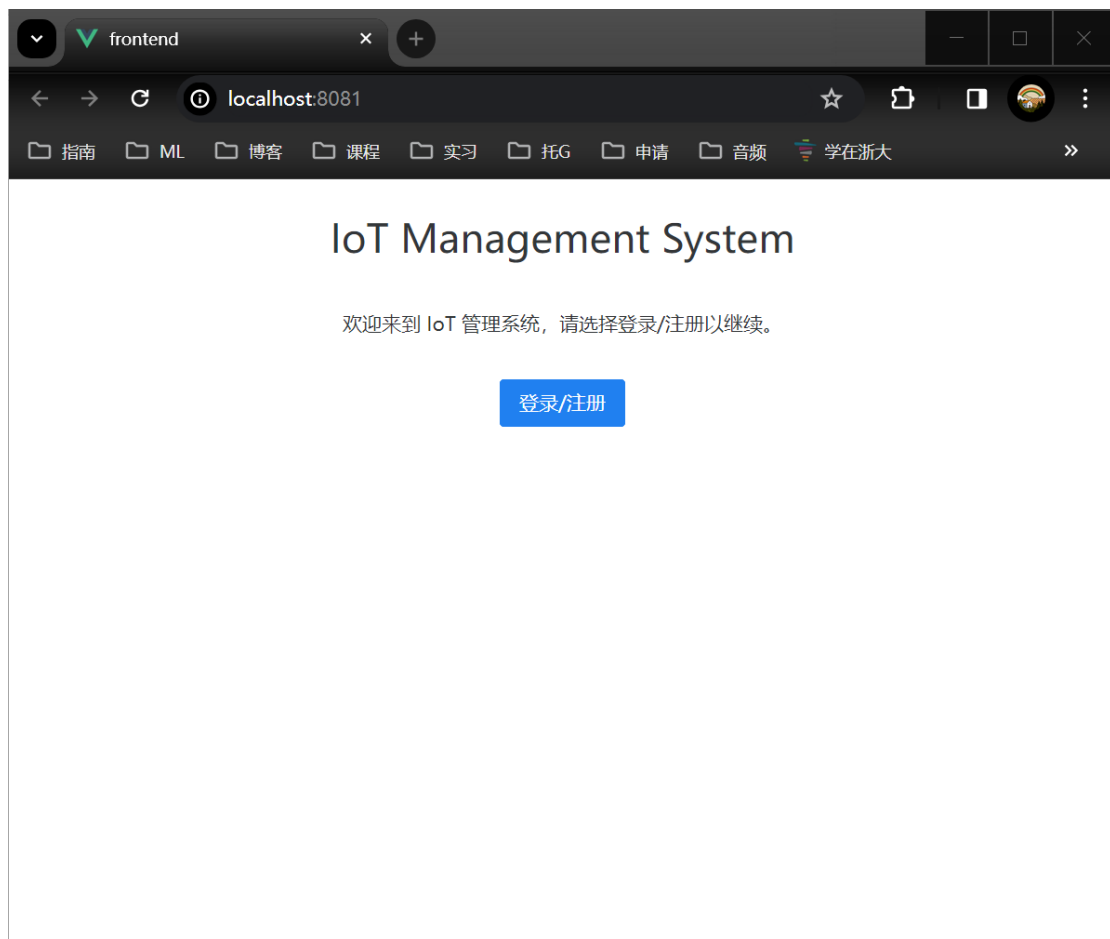
在安装了 Windows 10 操作系统的笔记本电脑上，启动 MySQL

和 Mosquitto 服务，编译 Golang 编写的后端并运行，用 `vue serve` 命令启动前端，通过 `chrome` 浏览器访问物联网管理平台。

3. 测试过程

3.1 引入页面

点击按钮后能正常跳转到登录与注册页面。



3.2 登录与注册

3.2.1 正常注册

由于软件刚刚启动，数据库里没有信息，所以需要先注册。邮箱

与电话是选填项。输入如下的信息，可以正常注册并进入工作台。

frontend

localhost:8081/login

指南ML博客课程实习托G申请音频学在浙大

登录注册

用户名

test

密码

.....

邮箱

1311766212@qq.com

电话

18106553735

注册

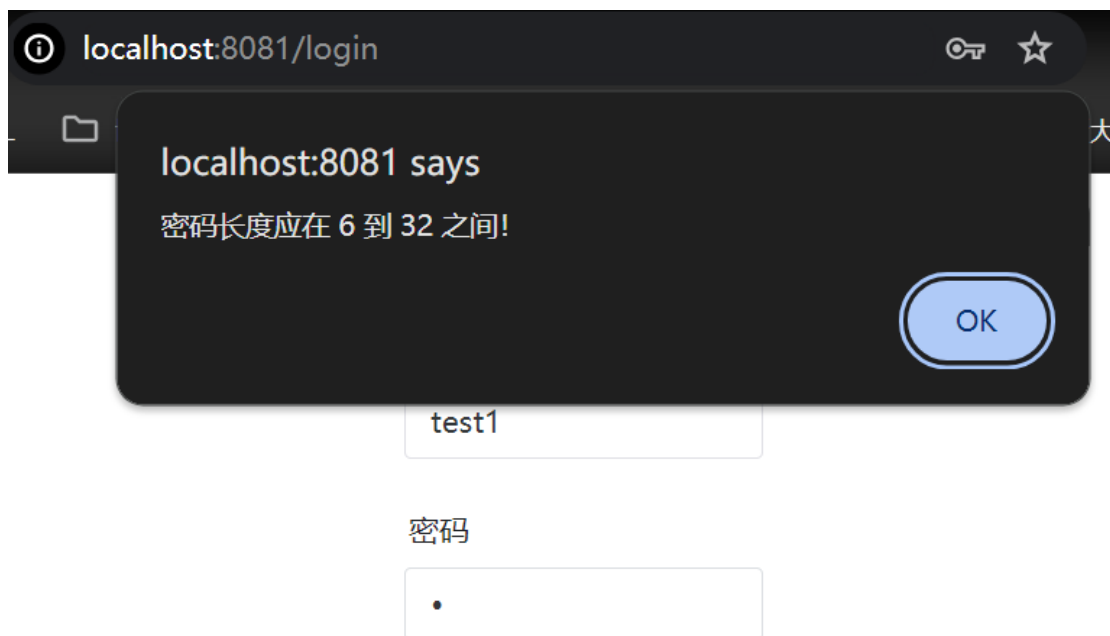


3.2.2 注册错误

用户名或密码为空时注册，会正常弹窗提示这两个字段不能为空。

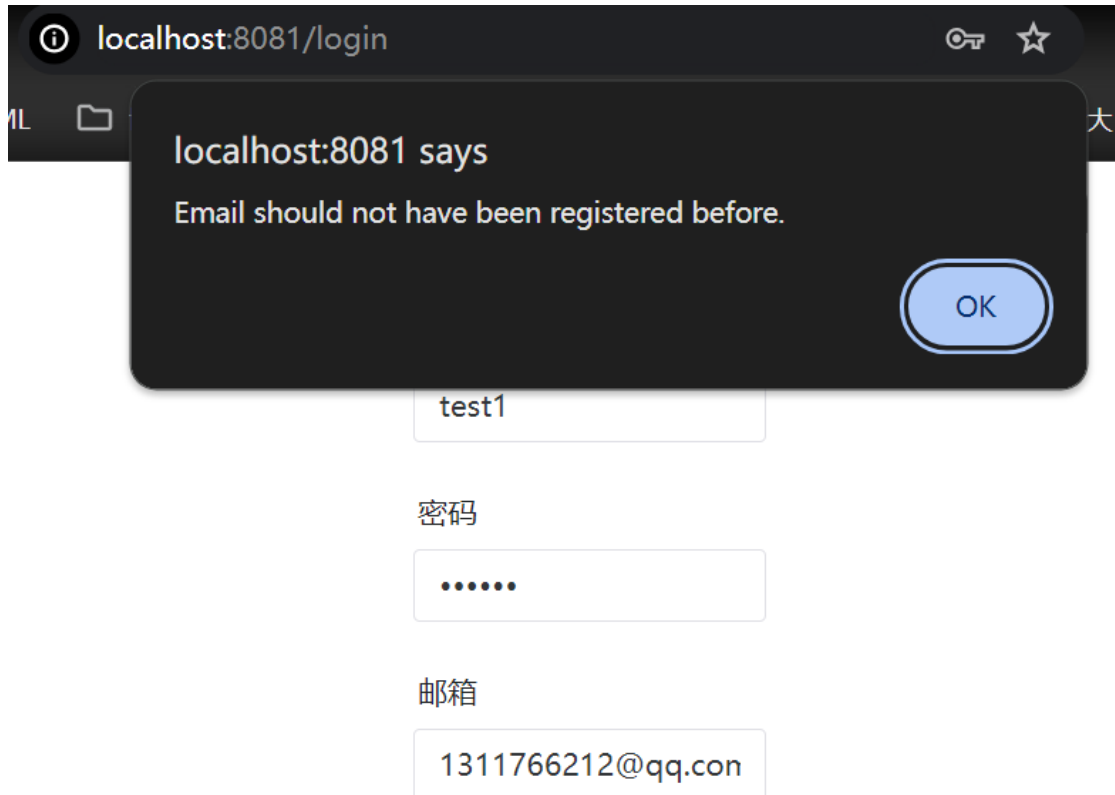


密码长度不合理时，会弹窗提示密码长度的要求。



邮箱或电话号码可以为空，但与已有用户重复时，会弹窗提示。这里用英文报错是因为查询邮箱、电话号码是否重复需要后端查询数据库再将可能出现的报错通知前端，而前后端用 ASCII 字符交流不易

出错。



为了方便后续测试，我们注册了这个 test1 账号，邮箱和电话号码都留空。

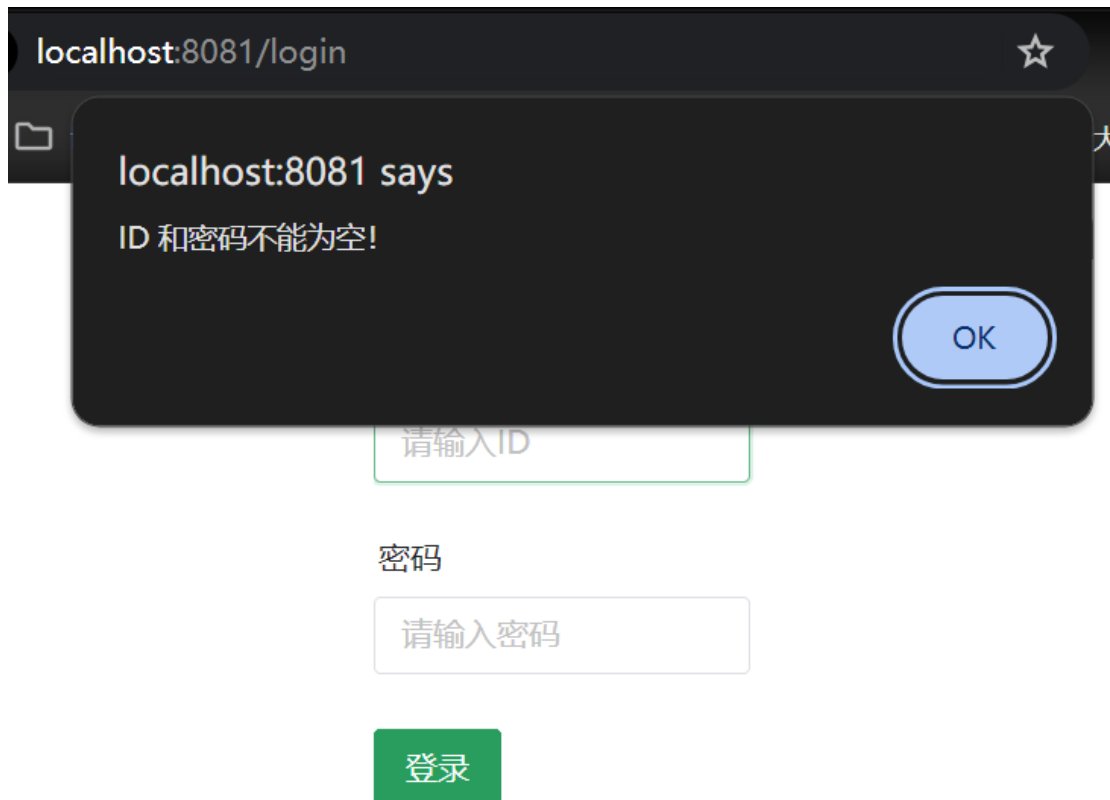
3.2.3 正常登录

在登录页面输入正确的 ID 和密码即可登录并跳转到工作台。



3.2.4 登录错误

ID 或密码为空、密码长度不正常时，前端会直接提示。



localhost:8081/login

localhost:8081 says
ID 和密码不能为空!

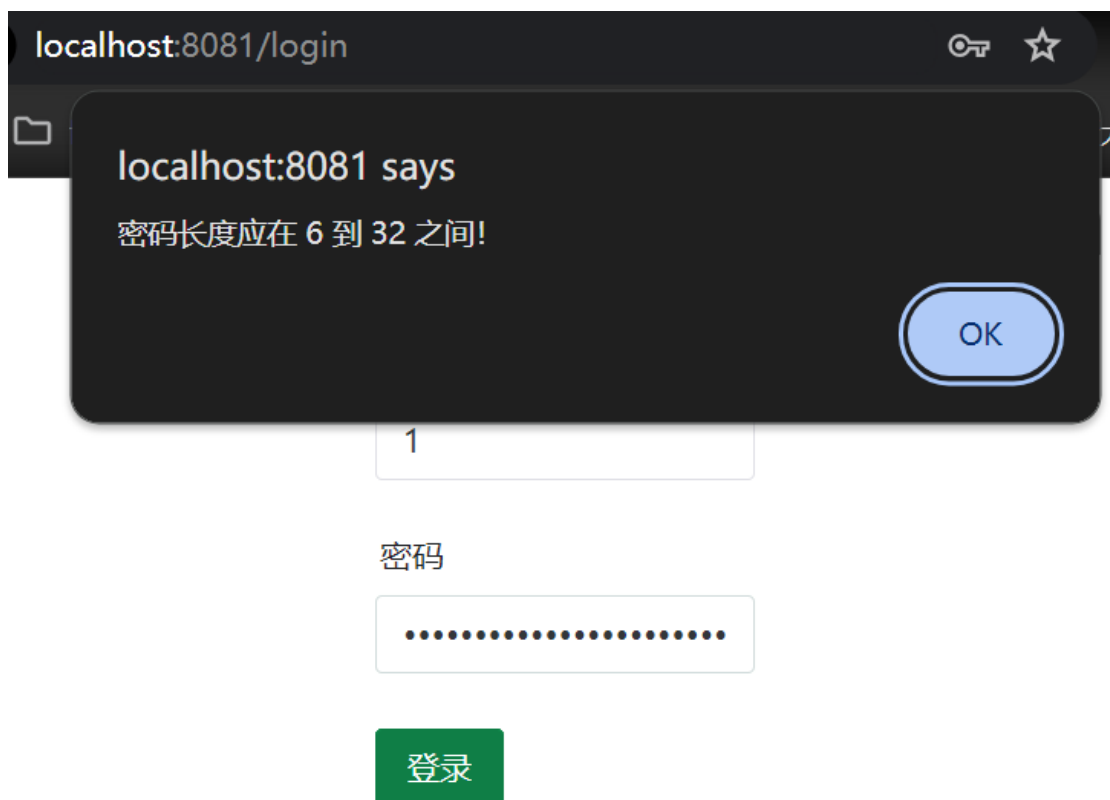
OK

请输入ID

密码

请输入密码

登录



localhost:8081/login

localhost:8081 says
密码长度应在 6 到 32 之间!

OK

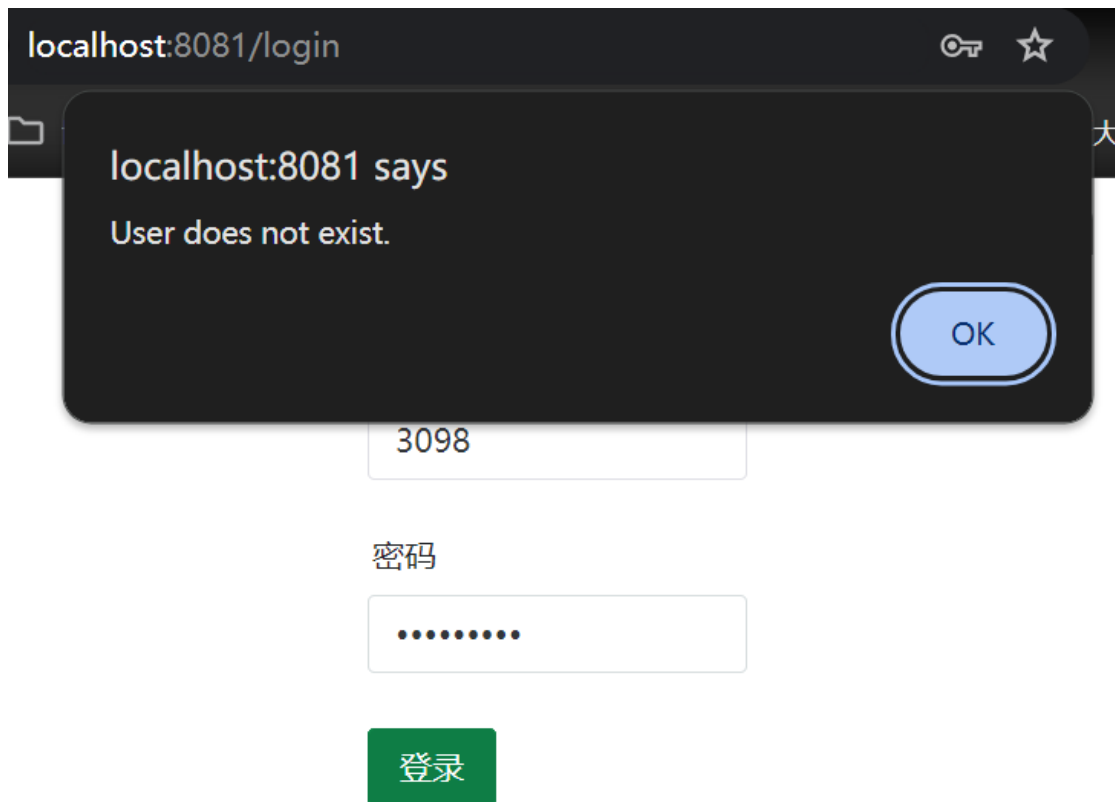
1

密码

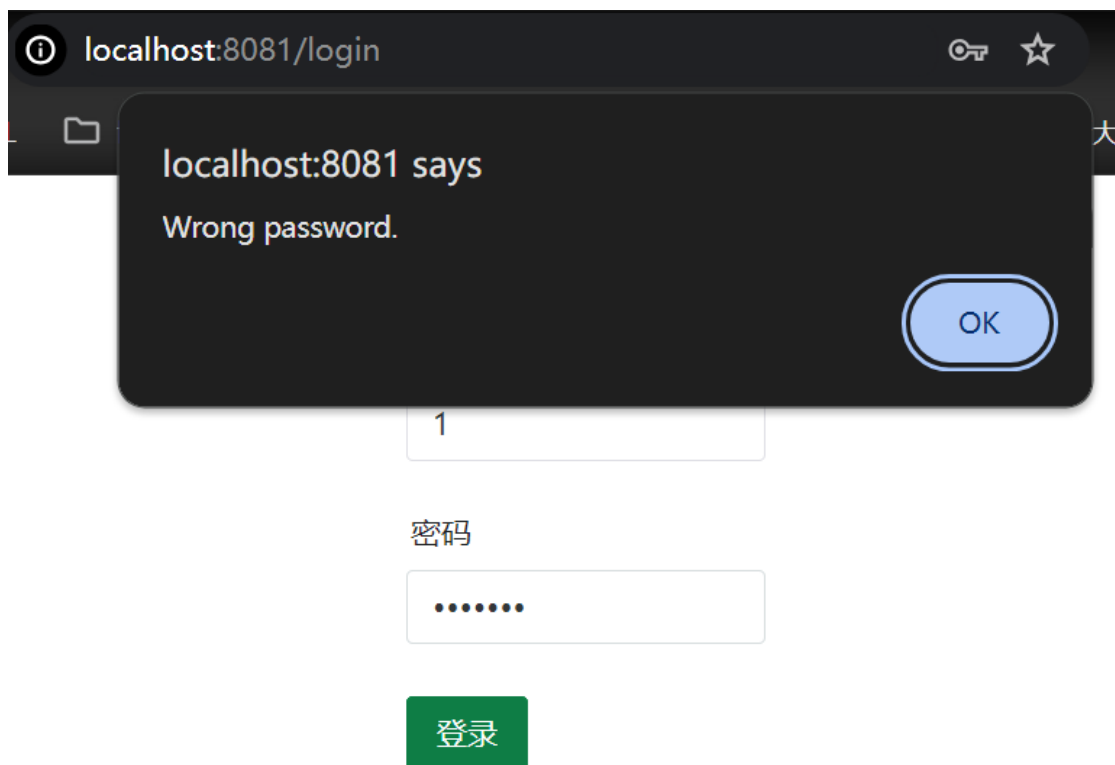
.....

登录

用户 ID 不存在时、用户存在但密码错误也有相应的提示。



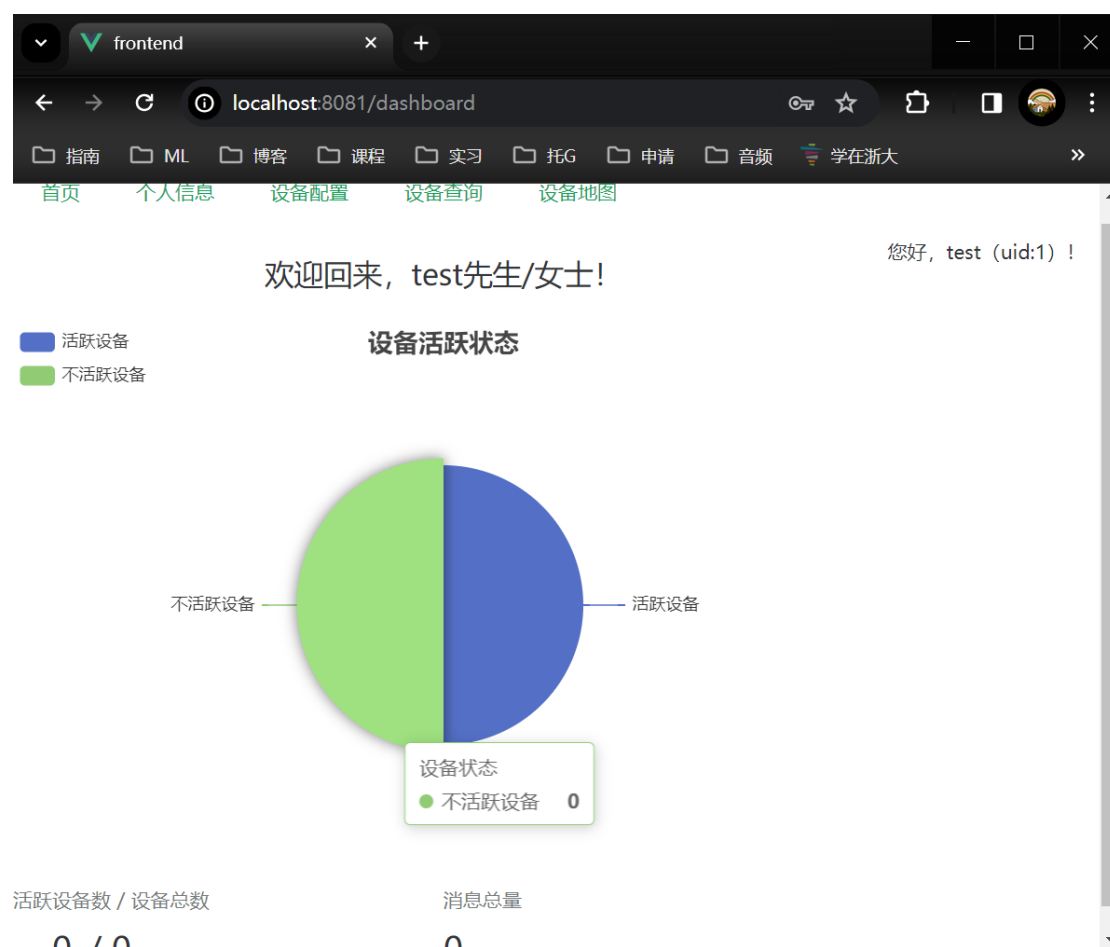
A screenshot of a web browser at the URL `localhost:8081/login`. A dark modal dialog box is displayed in the center, containing the text "localhost:8081 says" and "User does not exist." with an "OK" button. Below the dialog, the login form is visible, featuring a text input field containing the value "3098", a password input field with masked dots, and a green "登录" (Login) button.



A screenshot of the same web browser at `localhost:8081/login`. The modal dialog box now displays the text "localhost:8081 says" and "Wrong password." with an "OK" button. In the background, the login form shows the text input field containing the value "1", the password field is masked, and the green "登录" (Login) button remains.

3.3 工作台首页

登录后进入工作台首页。最上方为导航栏，可点击按钮跳转到其他界面，还显示了用户名和系统给用户分配的 ID。首页还显示了设备与消息的统计信息，方便用户登录后就能一目了然地看到自己设备的工作情况。

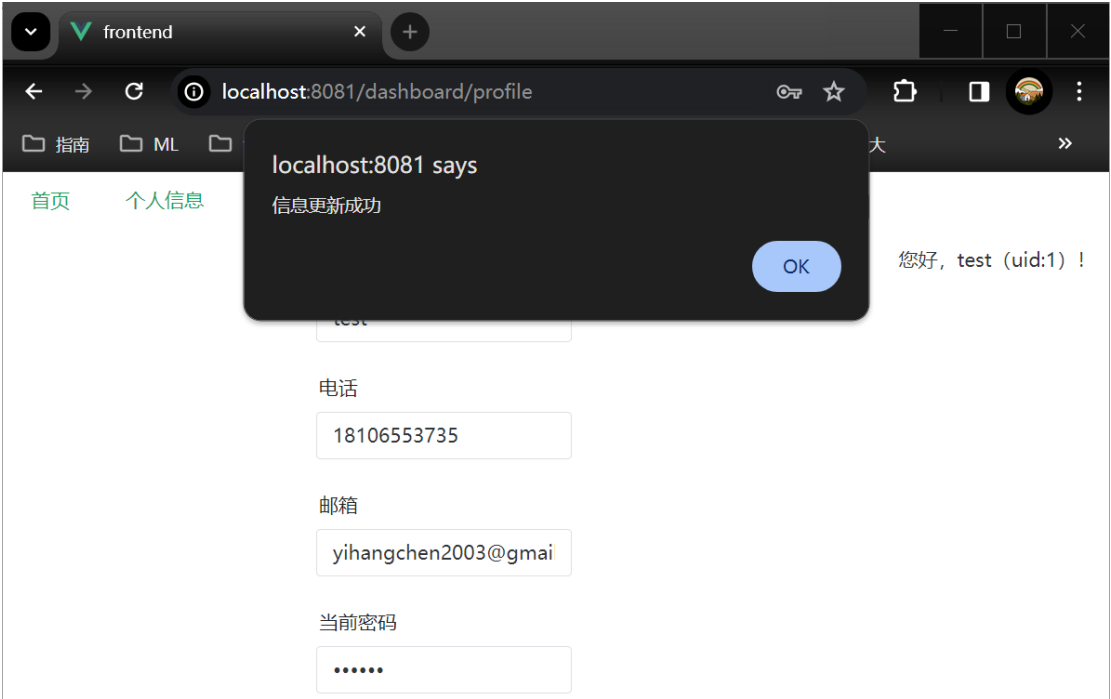
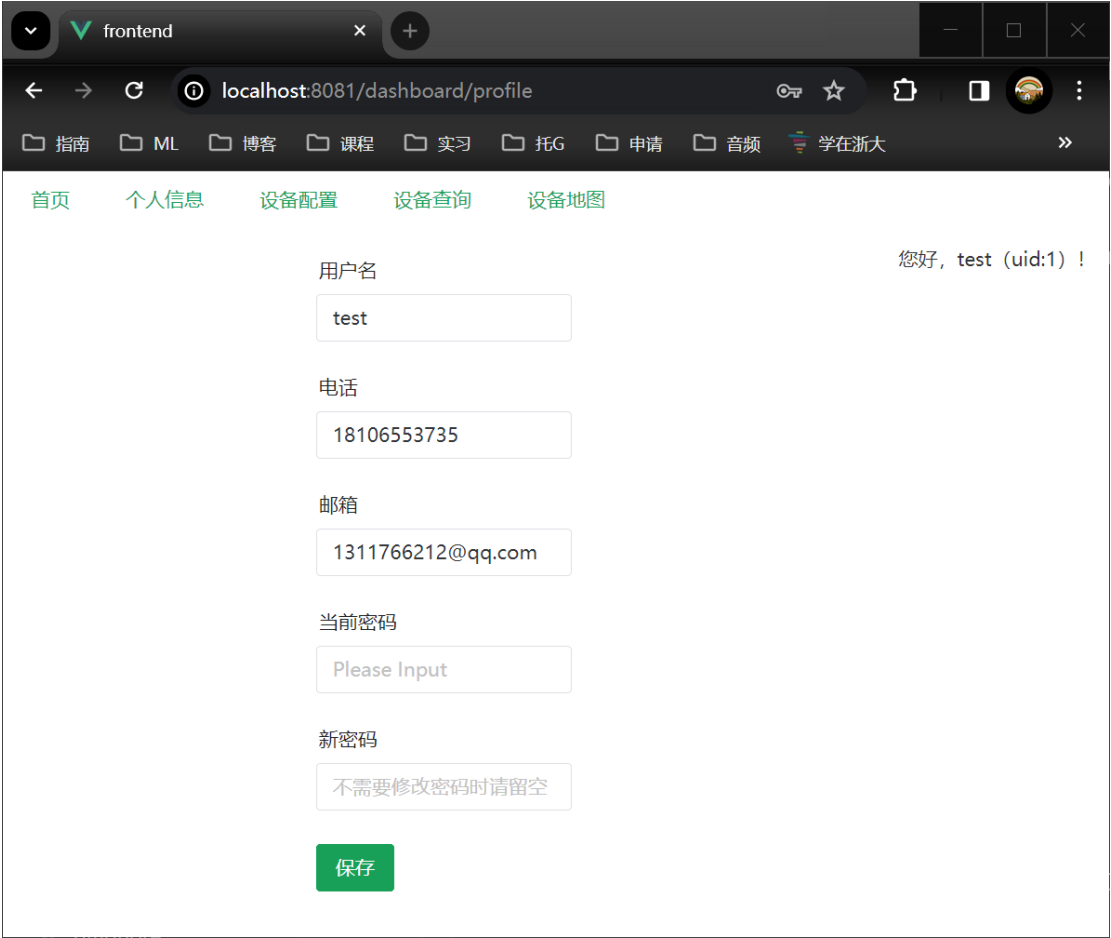


3.4 查看与修改用户信息

3.4.1 正常修改用户信息

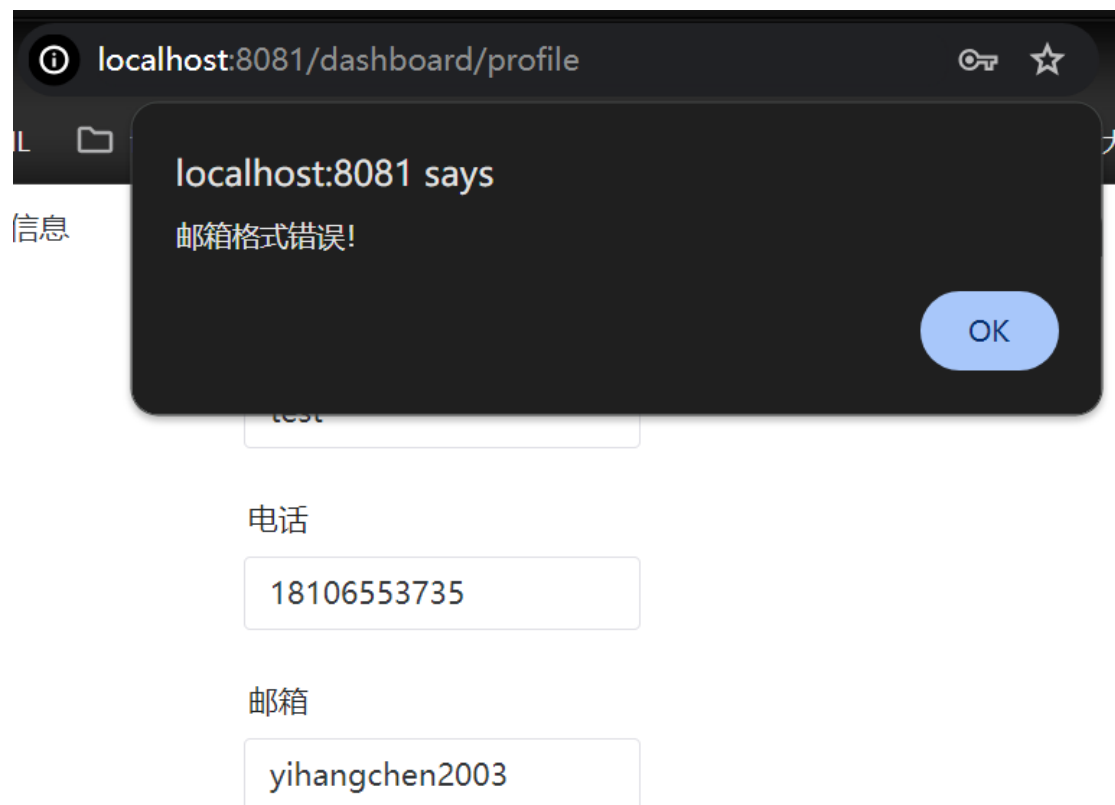
点击导航栏的“个人信息”按钮就会进入这个页面，输入框中默认显示除密码外的用户信息。修改其中的信息并填入当前的密码，就

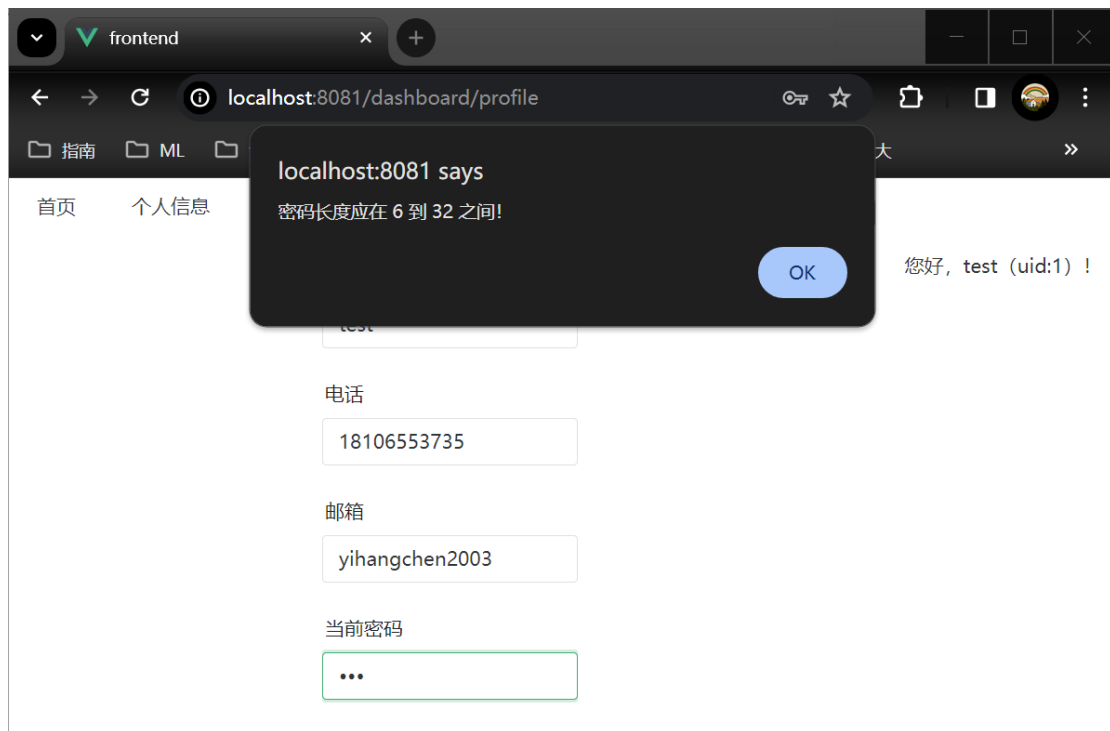
可以修改信息。



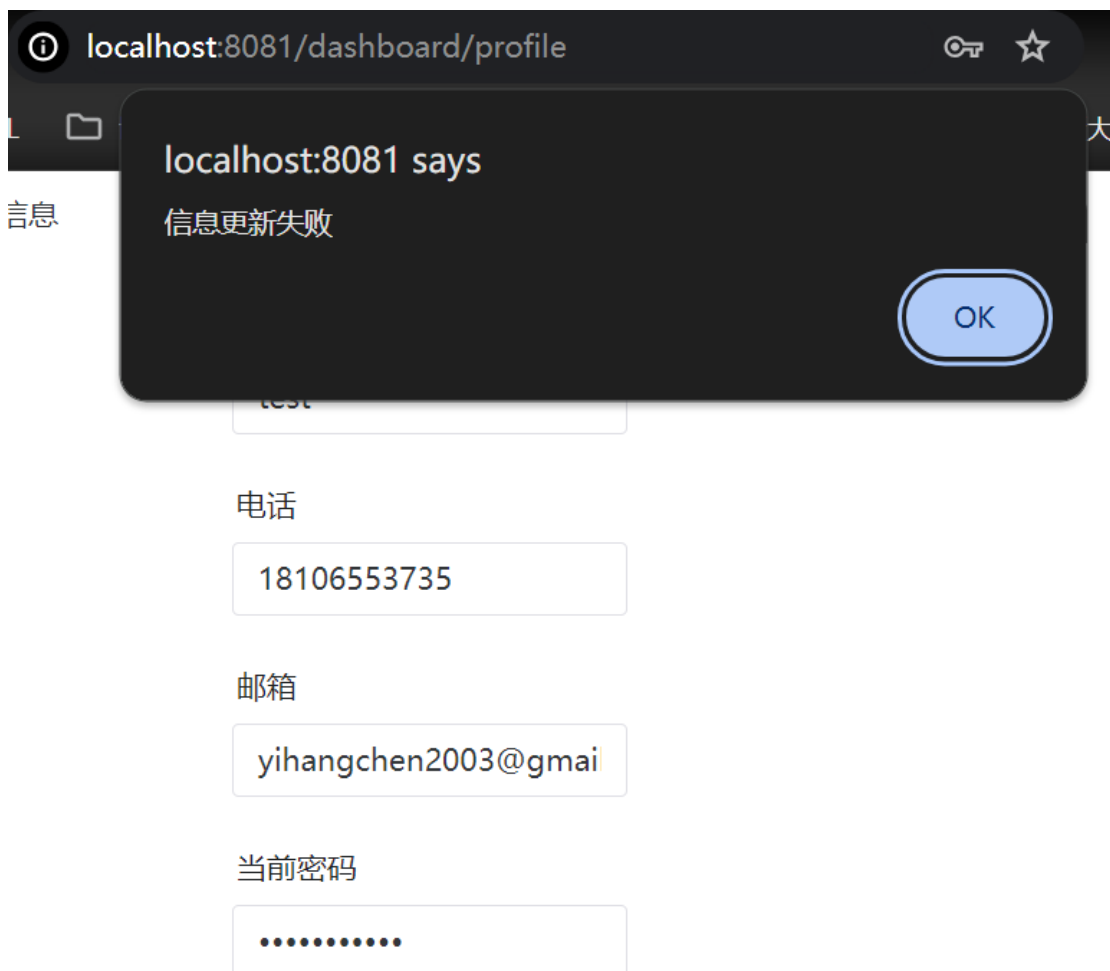
3.4.2 修改用户信息出错

与登录/注册界面同理，输入不合理的数据时会在前端直接报错。



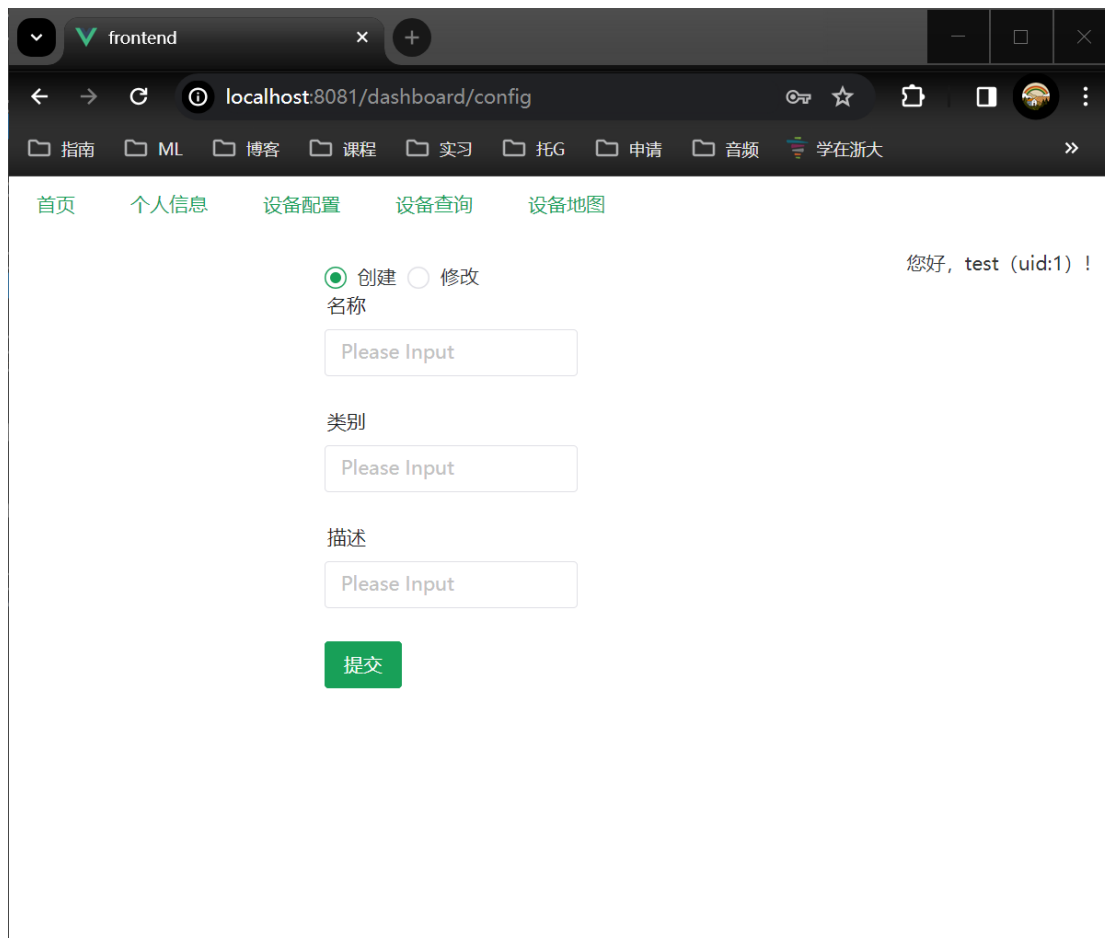


输入不正确的密码会接收到后端的报错。



3.5 创建与修改设备信息

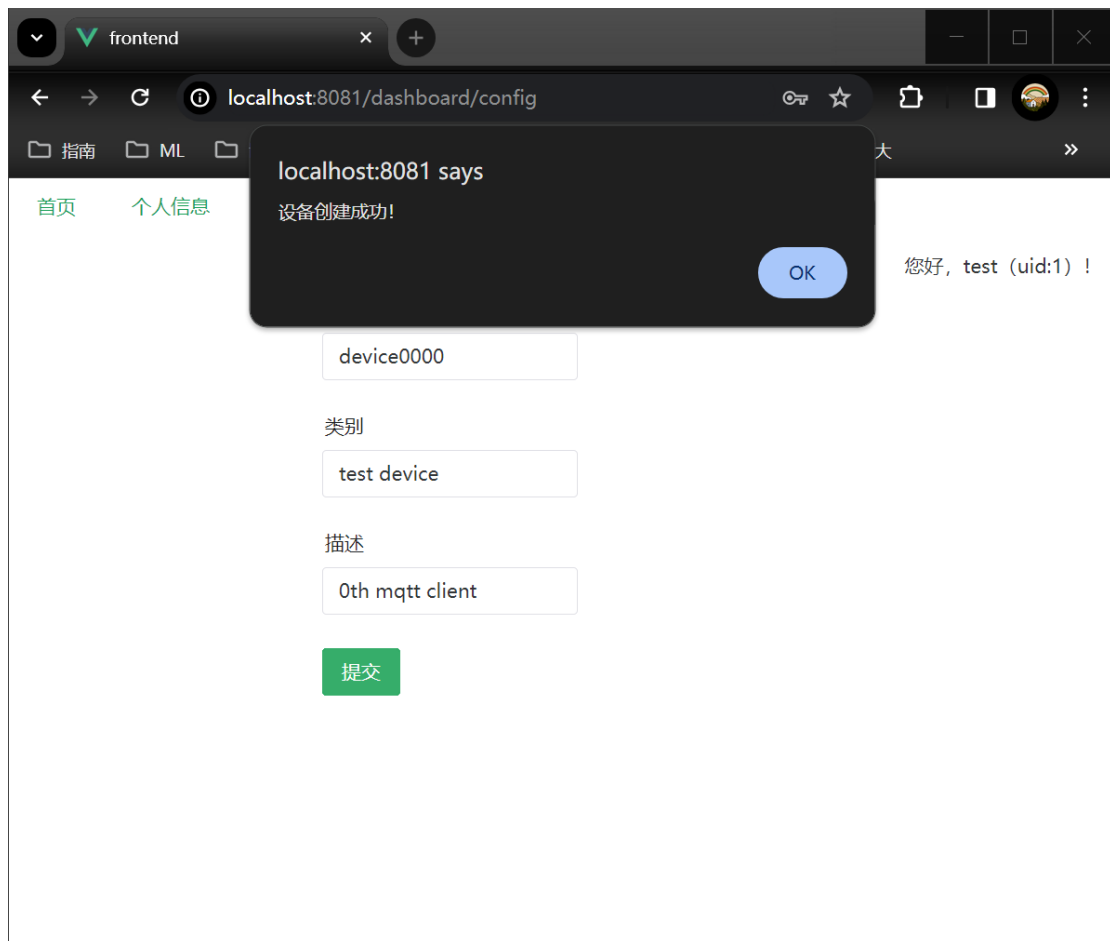
点击导航栏的“设备配置”按钮，进入配置页面。上方的“创建”和“修改”是一组单选按钮，用来决定是创建新设备还是修改已有设备的信息。



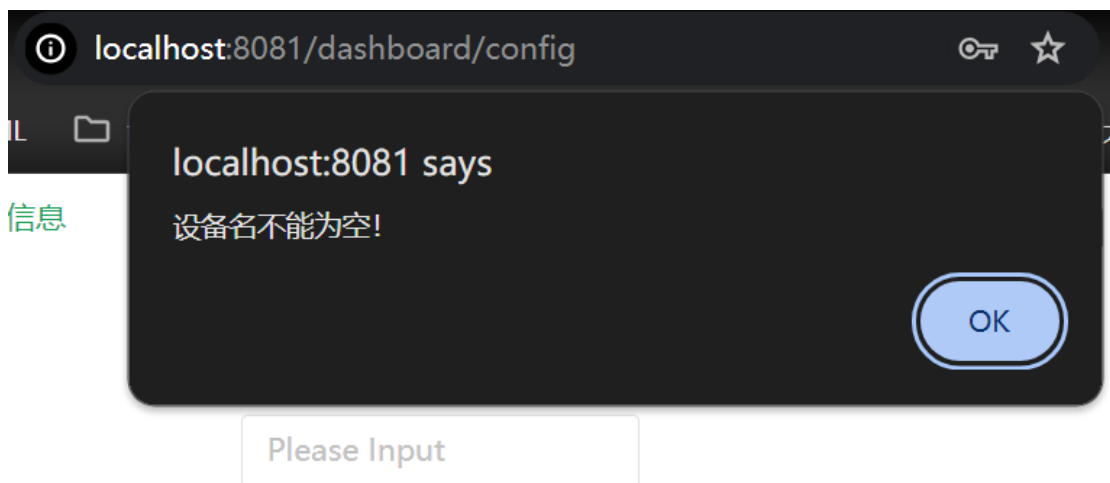
The screenshot displays a web application interface for device management. The browser window shows the URL `localhost:8081/dashboard/config`. The top navigation bar contains links for '指南' (Guide), 'ML', '博客' (Blog), '课程' (Course), '实习' (Internship), '托G', '申请' (Application), '音频' (Audio), and '学在浙大' (Study at ZJU). The main navigation bar includes '首页' (Home), '个人信息' (Personal Information), '设备配置' (Device Configuration), '设备查询' (Device Query), and '设备地图' (Device Map). The '设备配置' page features a form with two radio buttons: '创建' (Create) and '修改' (Modify). The '创建' button is selected. Below the radio buttons are three input fields labeled '名称' (Name), '类别' (Category), and '描述' (Description), each with a placeholder 'Please Input'. A green '提交' (Submit) button is located at the bottom of the form. A user greeting '您好, test (uid:1) !' is displayed in the top right corner.

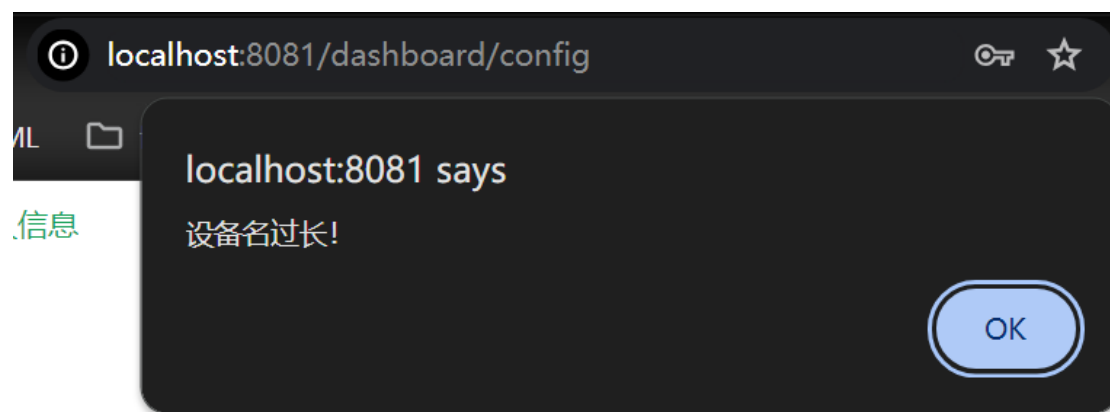
3.5.1 正常创建新设备信息

输入正常的设备名、类别名、描述并提交，即可创建新设备信息。其中类别名和描述可以留空。

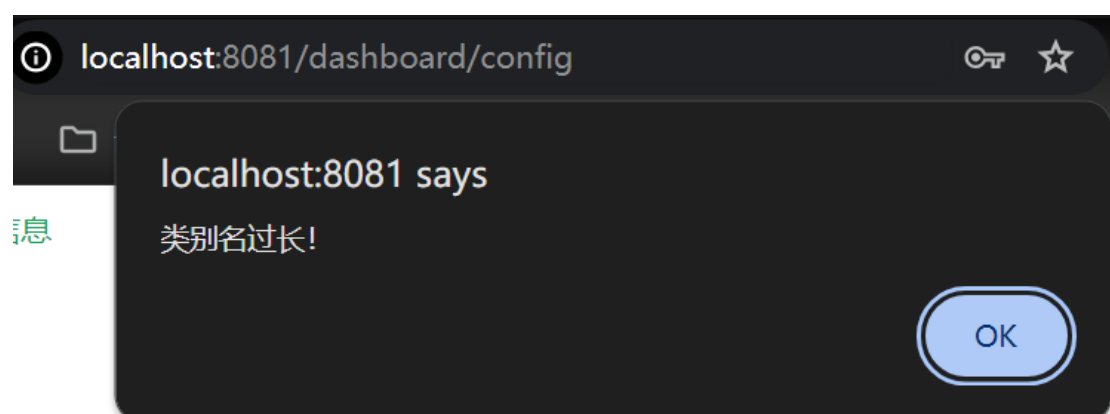


3.5.2 创建新设备信息错误





wozhendexuebuwanlesl



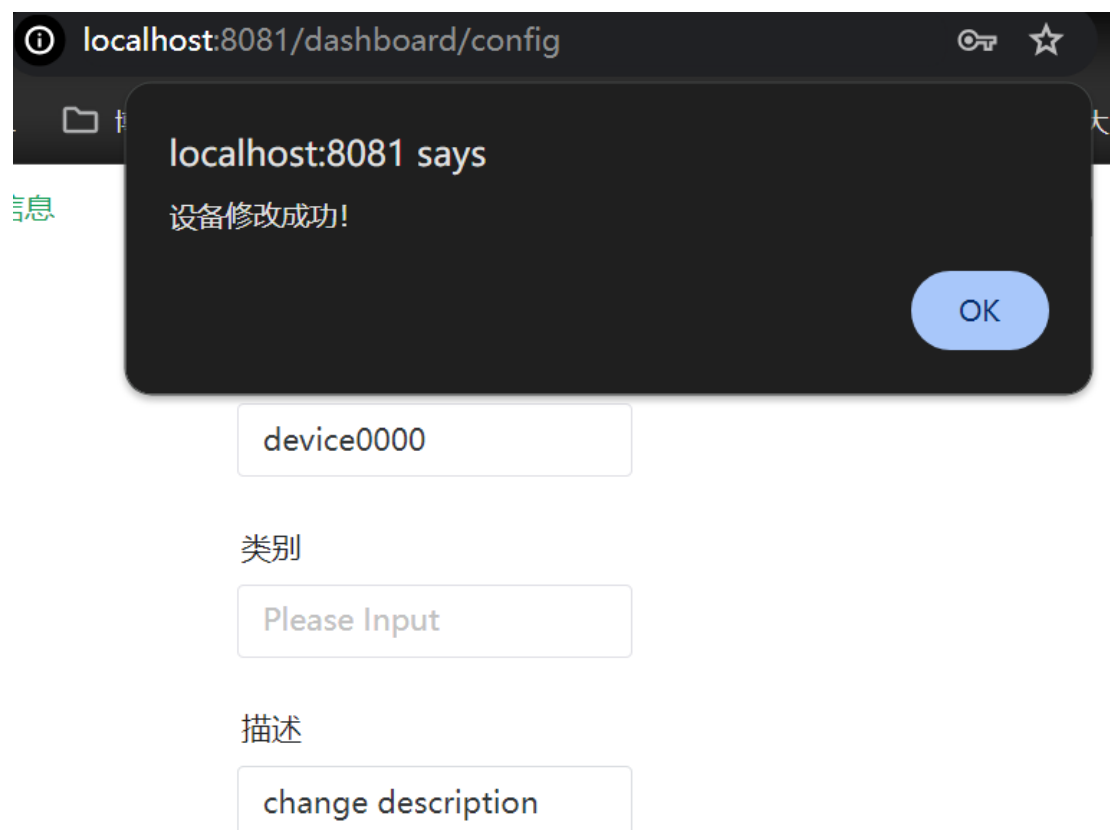
device0001

类别

wozhendexuebuwanlesl

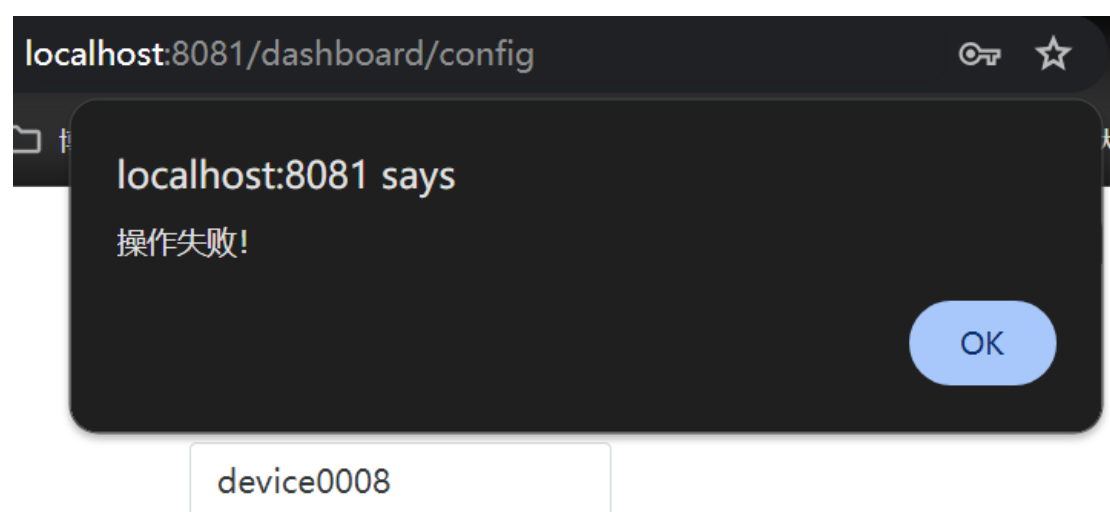
3.5.3 正常修改设备配置

在上方二选一选项里选择“修改”，输入已有设备的名字，即可重新设定其类别和描述。



3.5.4 修改设备配置错误

因为输入的名字、类别或描述信息不符合要求而修改失败的提示和创建设备时一样，因为共用了前端检测错误的代码。输入的设备名称在数据库中没有记录时，会弹窗提示“操作失败”。



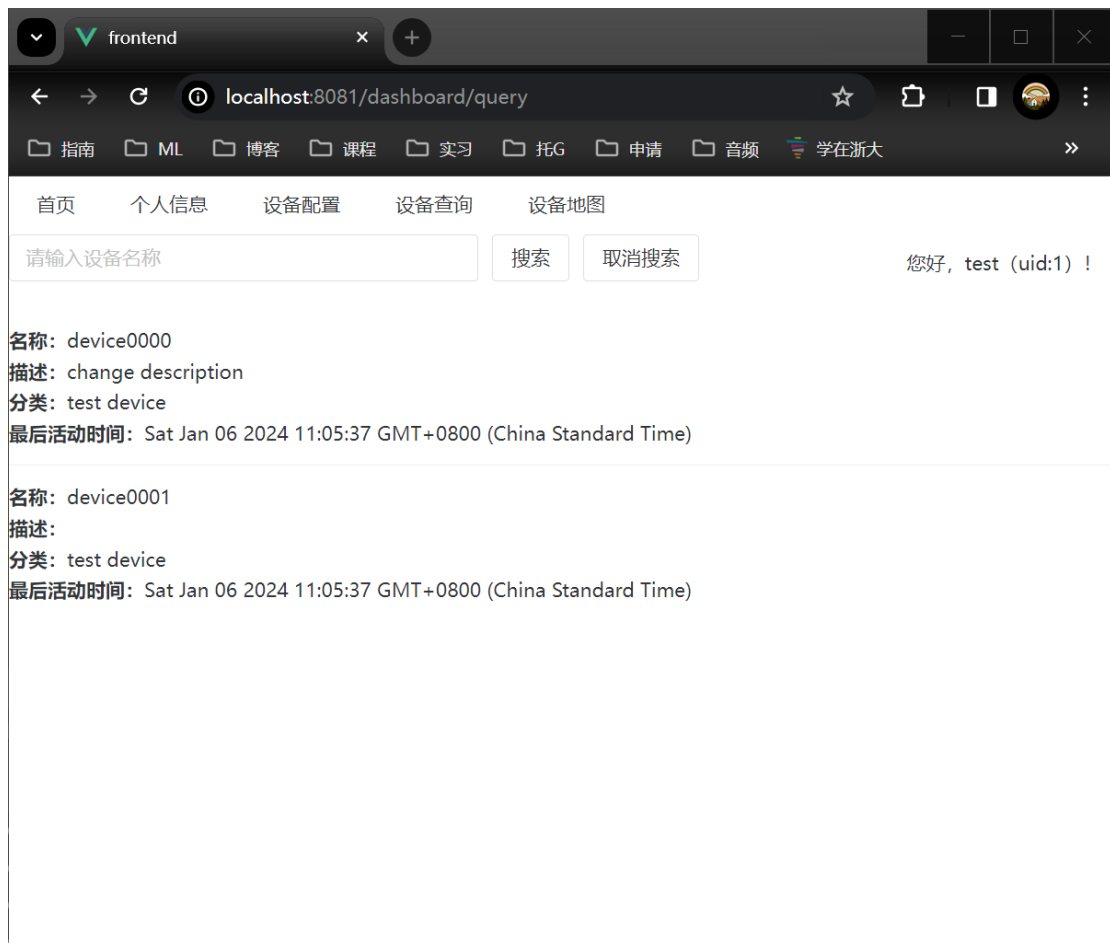
为了方便后续测试，我除了 device0000 又登记了 device0001，并且启动默认的 5 个 client 向 MQTT broker 发送数据。这里直接启动 client 即可，因为后端实现了一个 go routine 来接收 MQTT broker 发送来的消息并存入数据库，不需要调用其他服务。

再次进入工作台首页，可以看到由于 device0001 发送了消息，活跃设备和消息总量数量更新了。接下来，我们将测试设备查询和地图显示功能。

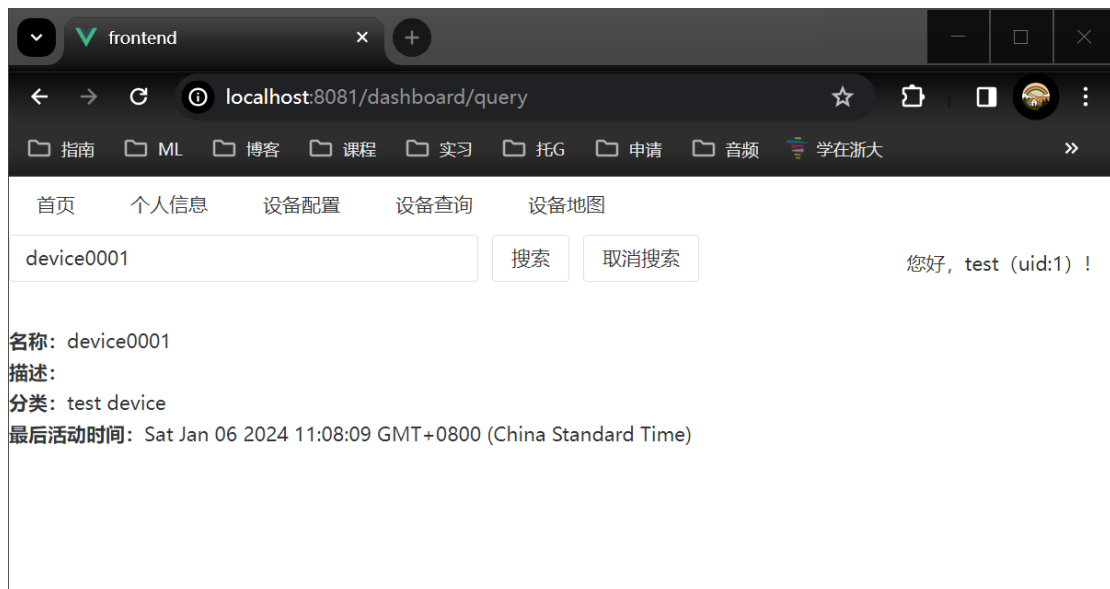


3.6 设备查询

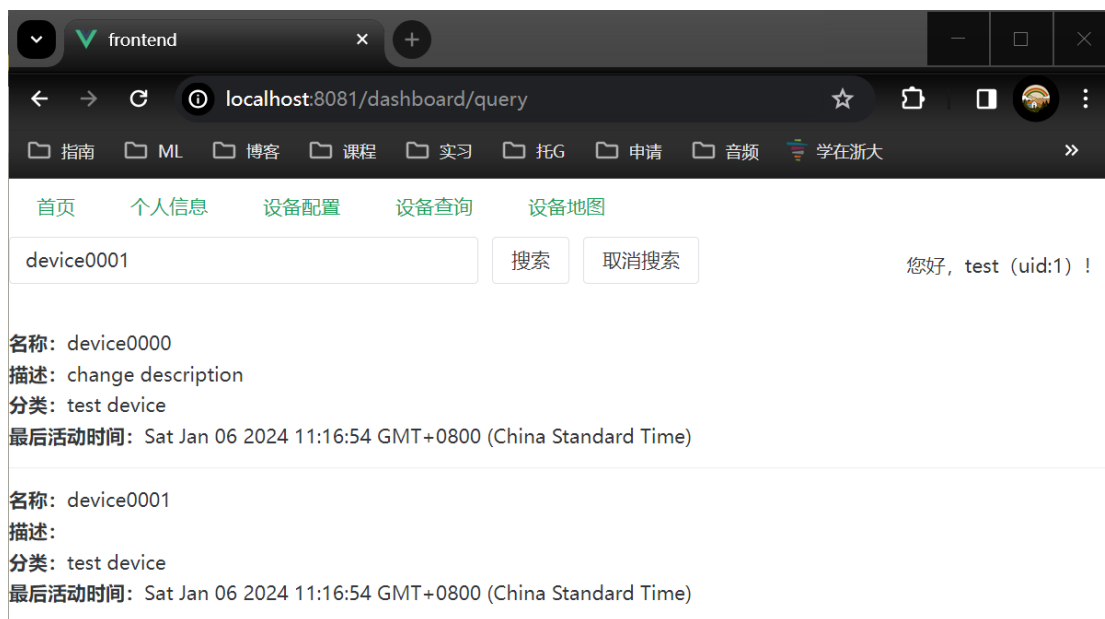
点击导航栏的设备查询按钮，进入设备查询页面。刚进入页面时，显示当前用户的所有设备。



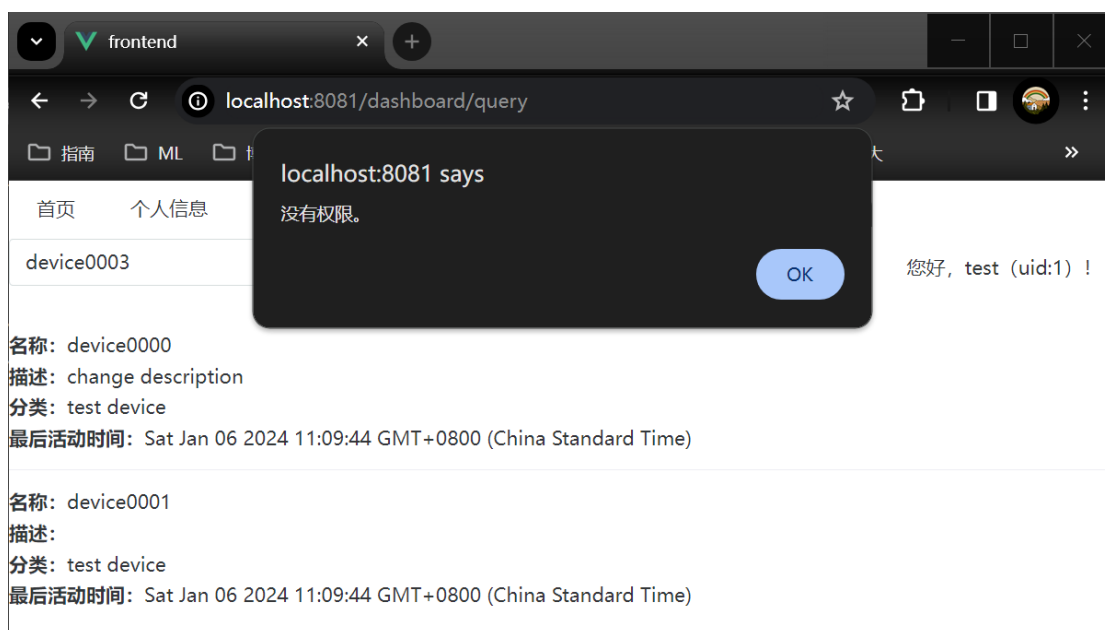
输入设备名称，点击搜索，若存在这一设备则显示其信息。



搜索完成后，点击“取消搜索”按钮，即可重新显示当前用户全部设备的信息。



若该设备不存在，或者存在但不属于该用户，则弹窗报错。

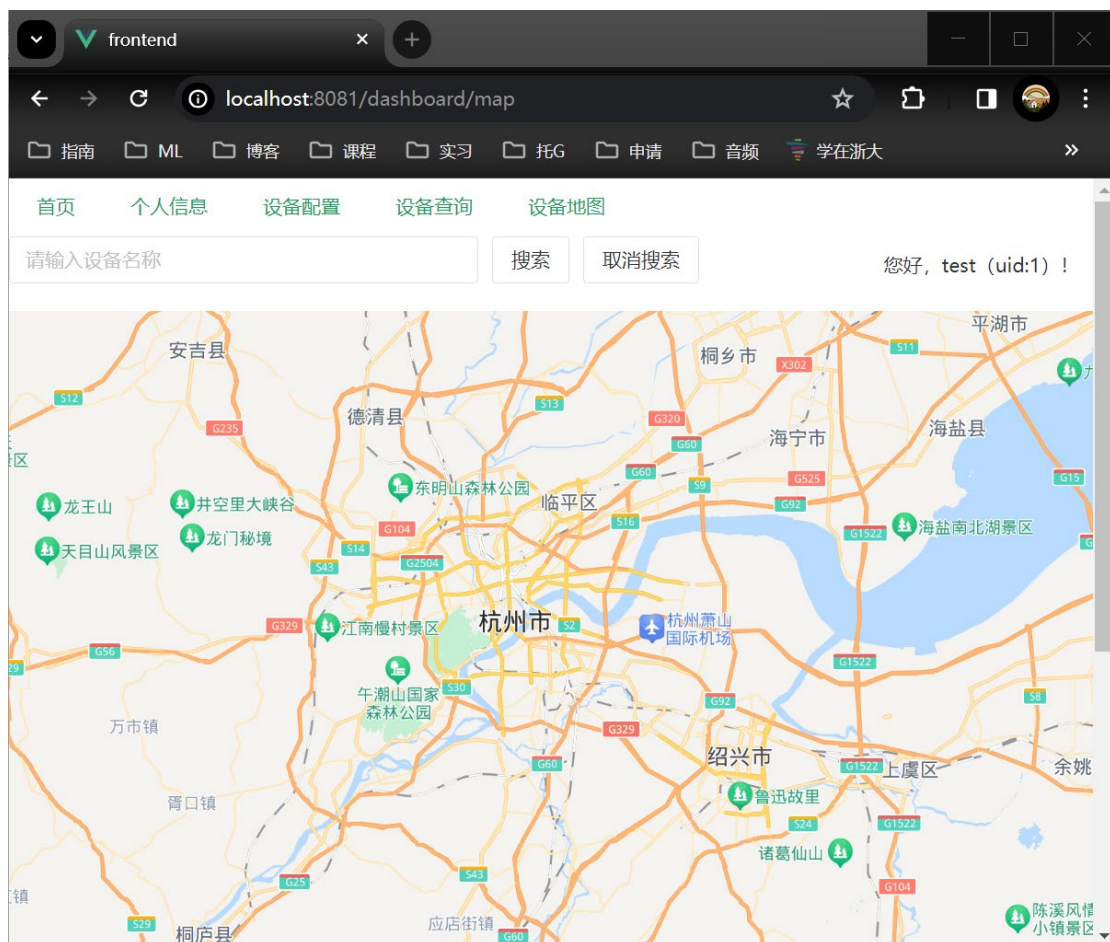


3.7 地图查询

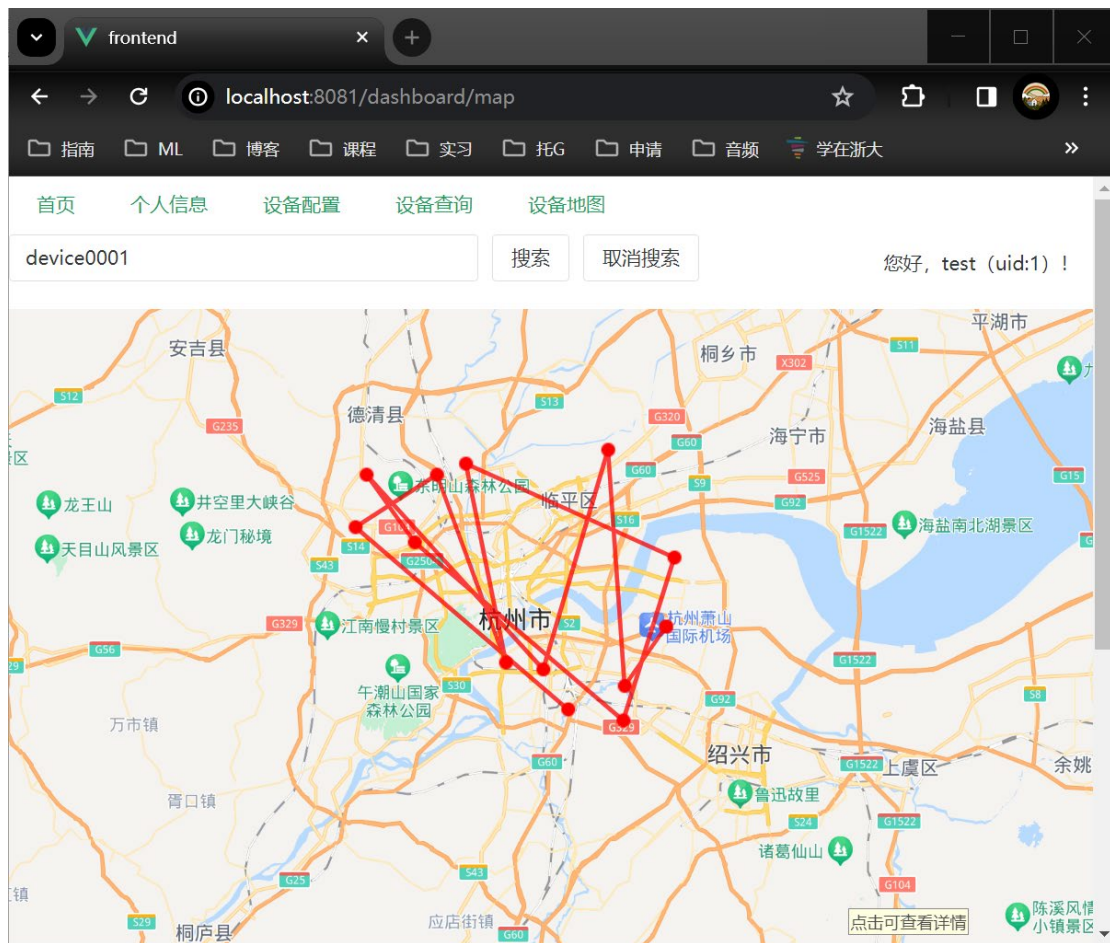
之前设计文档里说要用谷歌地图 API 来实现这一部分，一方面是因为谷歌地图有较成熟的前端 API，另一方面是因为它还能更好地适配外国的地图。但实际实现时发现在国内使用谷歌地图 API 有困难，

并且发现模拟设备发送消息的地址都在杭州，因此改成使用百度地图 API 了。百度地图 API 没有比较统一的 vue3 组件，有些功能的实现还存在问题，但可以方便地将地图中心设置为杭州等国内城市，学生还可以申请优惠，对国内开发者较为友好。

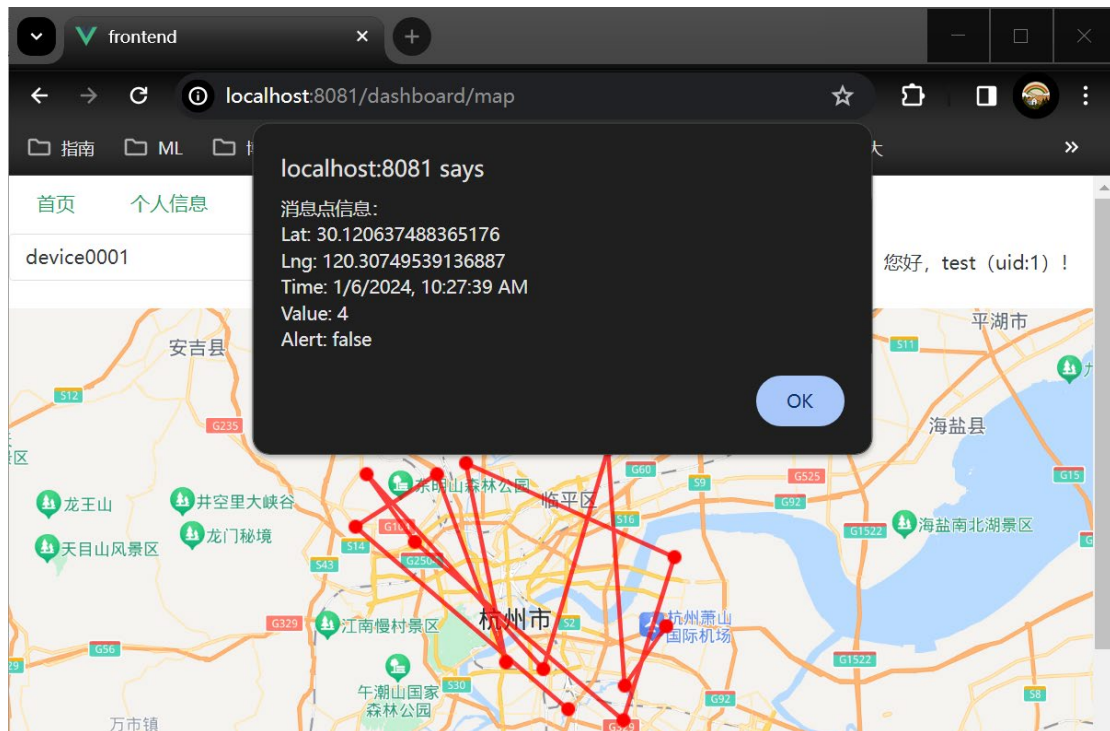
点击导航栏“设备地图”按钮进入地图页面。



输入要查询的设备名称，点击搜索，地图上就会用红线和红点来根据设备发送的消息显示其运动的路线和经过的地点。

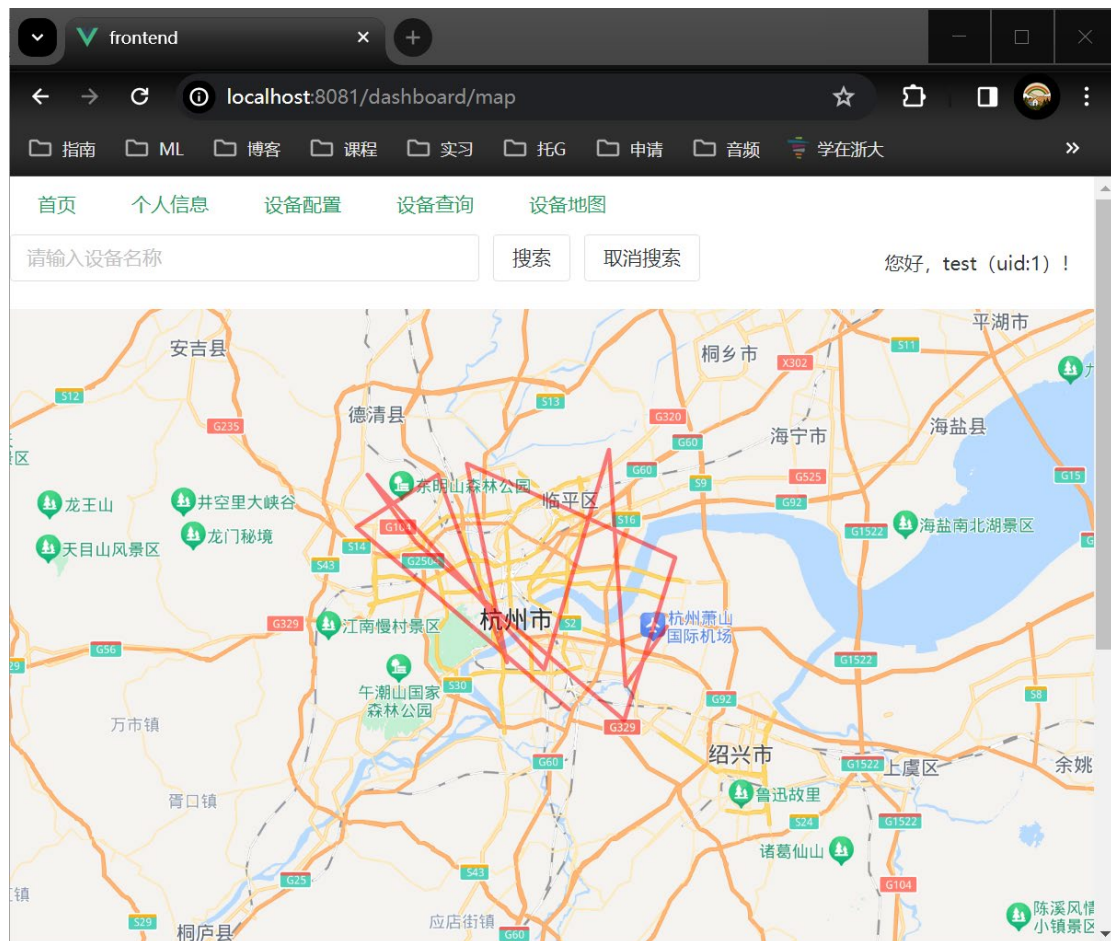


点击路径上的红点，就会显示设备在消息点处发送的信息。



点击取消搜索，在前端清空路径信息后，可以顺利清除消息点，

但之前的路径依然存在，效果如下：



尝试查阅文档解决问题，发现这主要是因为我使用的[百度地图组件库](#)不方便提供清除已经绘制到地图上的 `overlap` 的功能。

常见问题

- `BaiduMap` 组件容器本身是一个空的块级元素，如果容器不定义高度，百度地图将渲染在一个高度为 0 不可见的容器内。
- 没有设置 `center` 和 `zoom` 属性的地图组件是不进行地图渲染的。当 `center` 属性为合法地名字符串时例外，因为百度地图会根据地名自动调整 `zoom` 的值。
- 由于百度地图 JS API 只有 JSONP 一种加载方式，因此 `BaiduMap` 组件及其所有子组件的渲染只能是异步的。因此，请使用在组件的 `ready` 事件来执行地图 API 加载完毕后才能执行的代码，不要试图在 vue 自身的生命周期中调用 `BMap` 类，更不要在这些时机修改 `model` 层。

4. 测试结果总结

本次作业要求实现的功能有：

- 搭建 MQTT 服务器，接收模拟设备发送的消息
- 实现用户登录、注册
- 在首页用图表展示用户设备的统计信息
- 配置用户的设备，包括创建新设备和修改旧设备的配置
- 查询用户的设备
- 提供地图页面来可视化设备在不同地点发送的信息，可区分普通和警告信息（点击消息点后弹窗）

经过上述测试，以上基本功能均能实现。