使用手册

1 运行环境

1.1 软件环境

- Node.js
- python 3.11
- mySQL
- mosquitto
- Vue
- Flask

1.2 硬件环境

- CPU：主频大于 2 GHz
- 硬盘： 硬盘容量大于 200 GB、硬件转速大于等于 5400 转/分钟
- 其他硬件满足正常的使用需求即可

2 运行方法

- 前端进入 `code/frontend/client` 后运行 `npm run dev` 命令即可运行
- 后端进入 `code/backend` 后运行 `python app.py` 命令即可运行
- 再开启一个终端，运行 `code/mqtt/mqttServer.py` 文件
- 对于模拟家具信号的发送可通过更改 `code/mqtt/mqttClient.py` 文件中 `send` 变量的内容并运行该文件实现
- 对于数据库所用到的相关建表 sql 语句在 `code/mysql` 文件夹中可以找到

3 项目介绍

3.1 项目背景

- 智能家居是在互联网影响之下物联化的体现。智能家居通过物联网技术将家中的各种设备连接到一起，提供家电控制、照明控制、电话远程控制、室内外遥控、防盗报警、环境监测等等功能。智能家居的概念起源很早，到1983年美国联合科技公司将建筑设备信息化、整合化概念应用于美国康涅狄格州哈特佛市的 CityPlaceBuilding 时，才出现了首栋"智能型建筑"，从此揭开了全世界争相建造智能家居的序幕
- 本网站为一个智能家居管理网站，基于 B/S 结构设计，实现了智能家居管理的基本功能以及模拟家居终端信息的发送和接收功能。

3.2 项目功能

3.2.1 用户模块

- 用户注册
- 用户登录
- 用户信息查看
- 用户信息修改

3.2.2 家居管理模块

- 场景创建与查看
- 设备创建与查看
- 设备详细信息接收展示

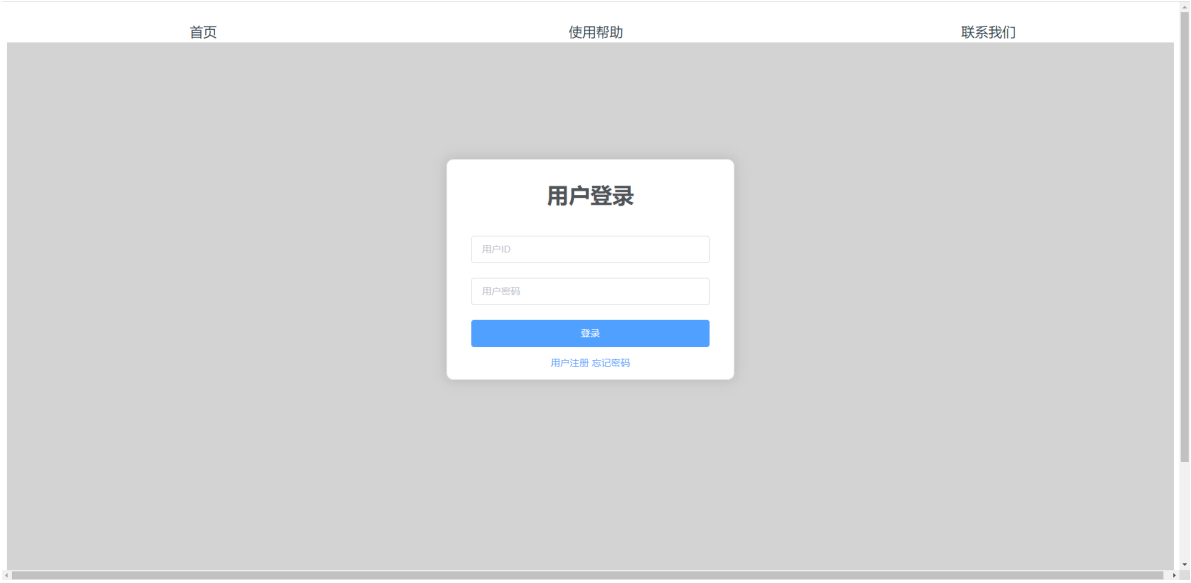
3.2.3 具体功能

- 用户注册时用户名、密码以及手机的格式要求
- 用户登录后可以创建场所，然后在场所中创建智能设备
- 支持四种设备：灯、开关、传感器、门锁
- 提供列表信息查看设备信息、设备状态以及上报信息
- 可以在房间户型图上摆放设备

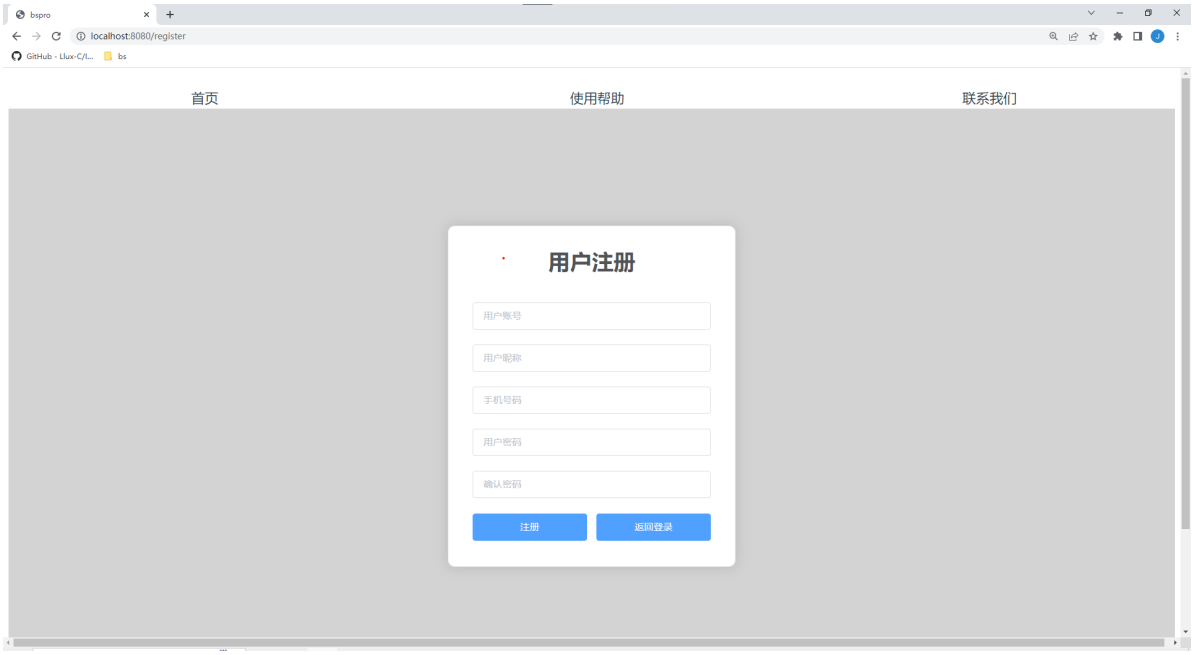
4 操作指南

4.1 用户注册与登录

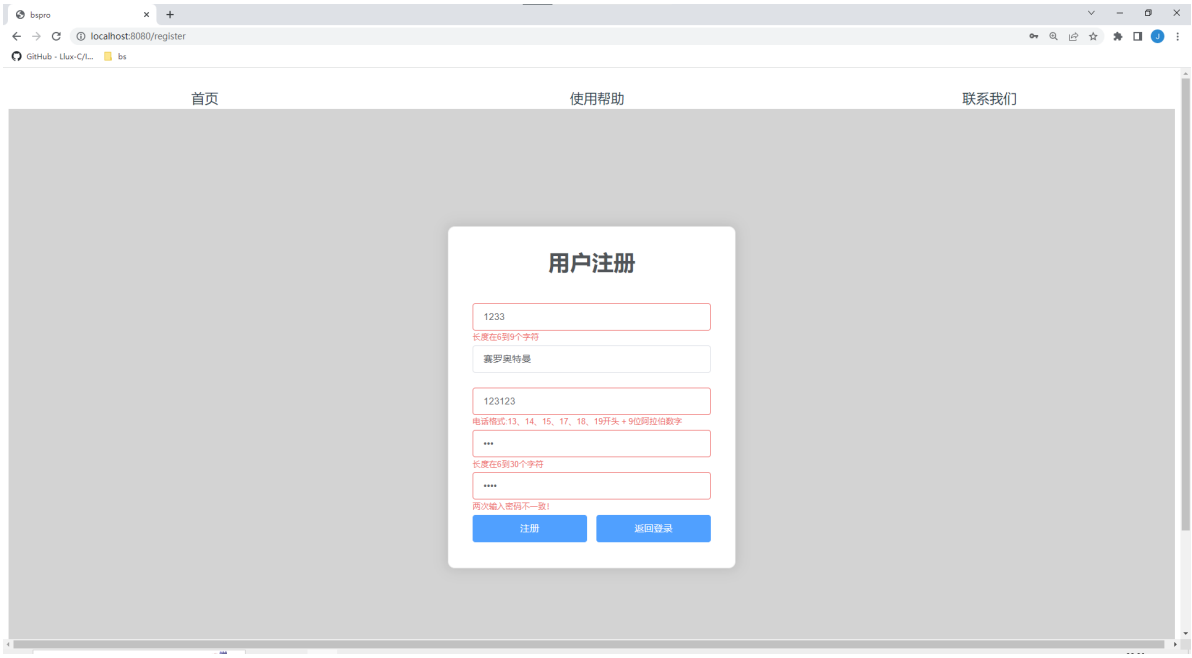
- 进入主页后，即可输入已经创建好的账号密码进行登录。若没有账号，可点击下方用户注册按钮跳转注册页面进行注册。



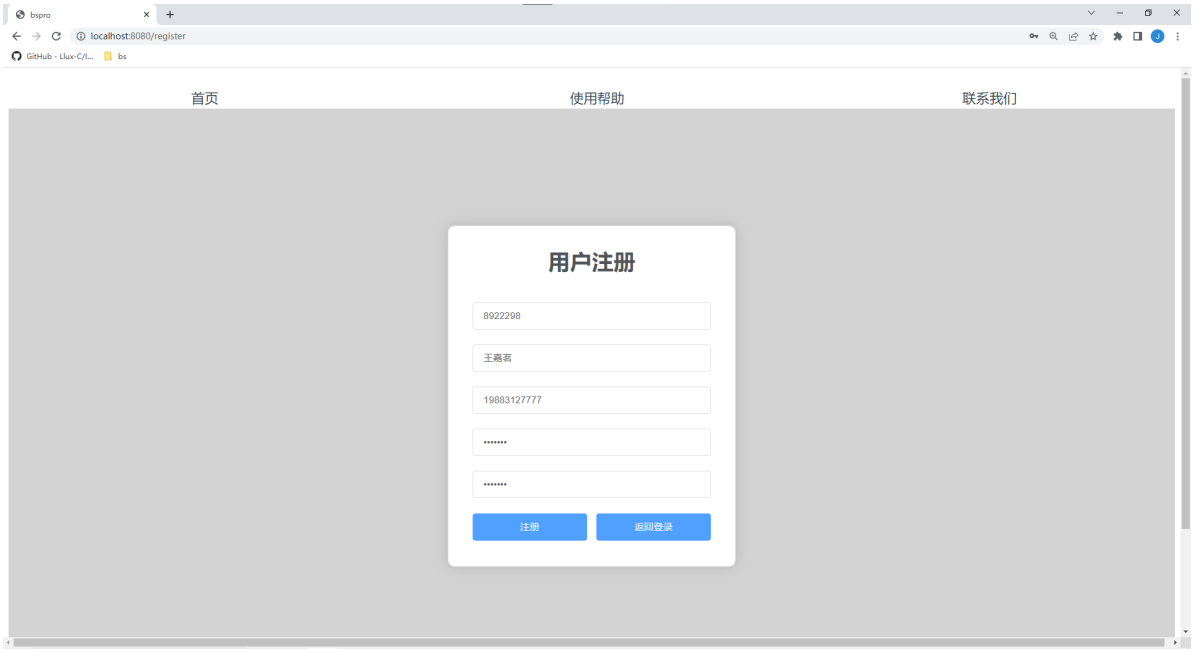
- 进入注册界面按照要求填写各项信息，点击注册按钮即可完成注册。



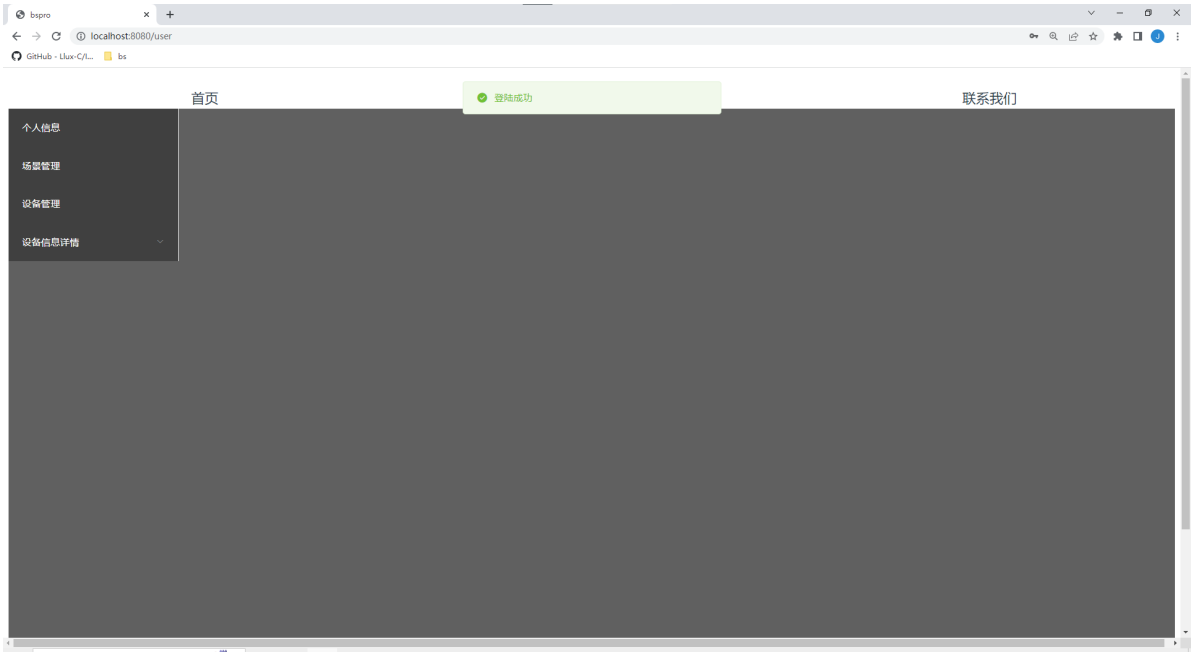
- 若输入信息不符合格式要求，则无法注册。



- 注册信息格式满足要求

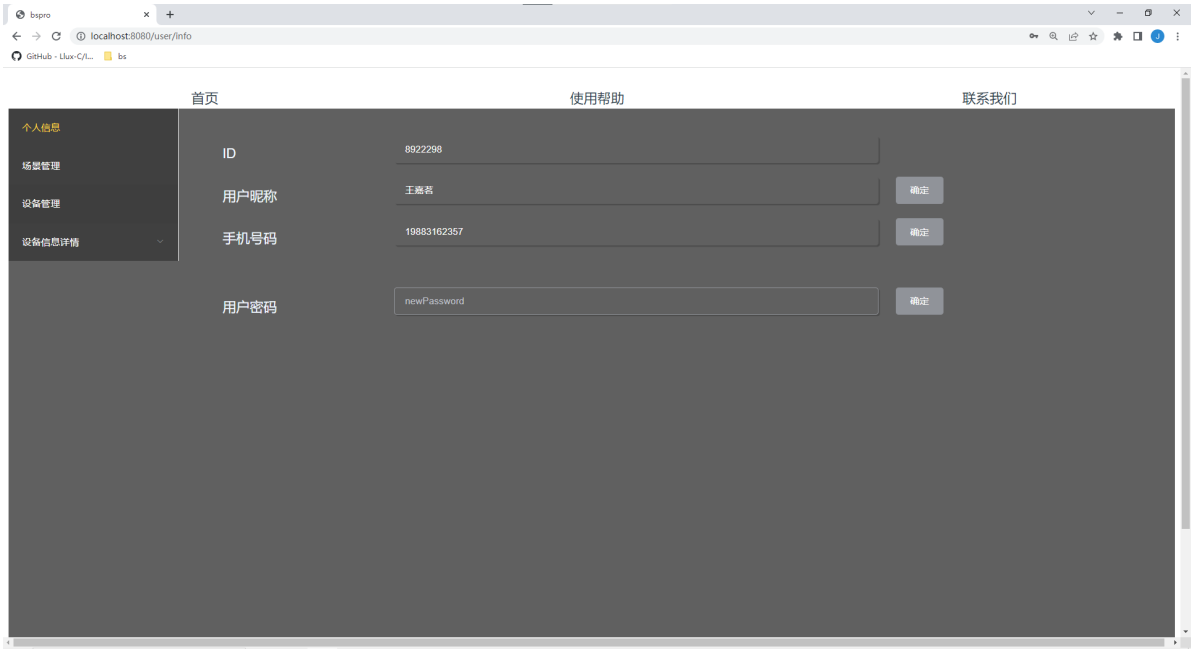


- 登录之后进入登陆后界面

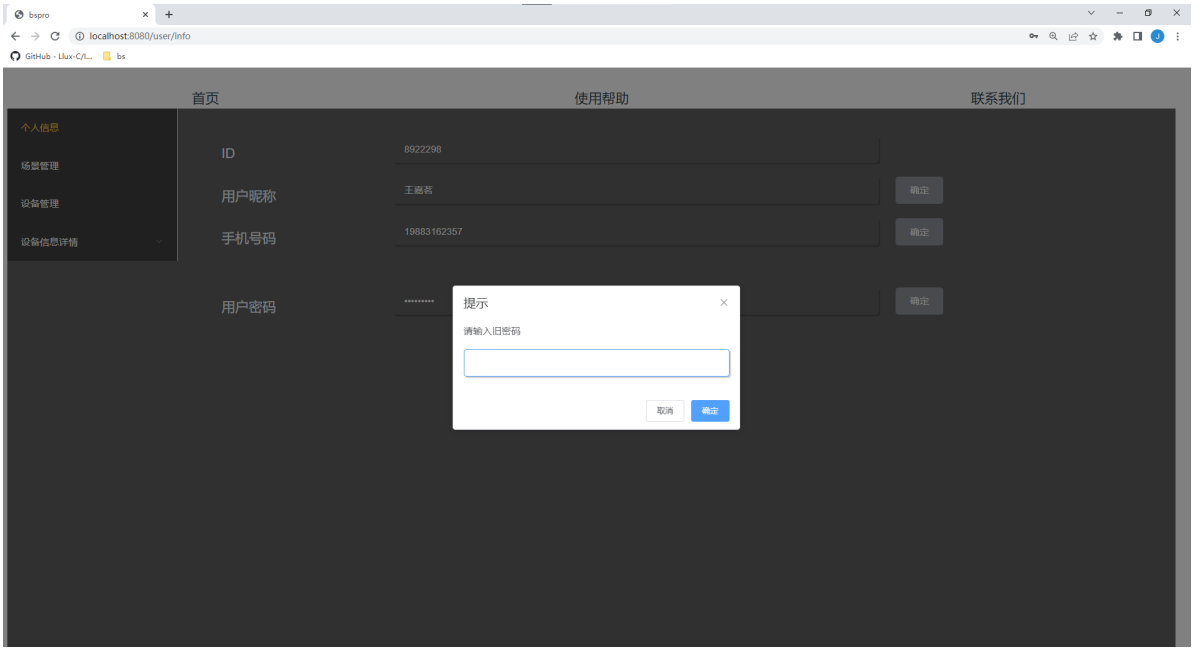


4.2 用户信息修改

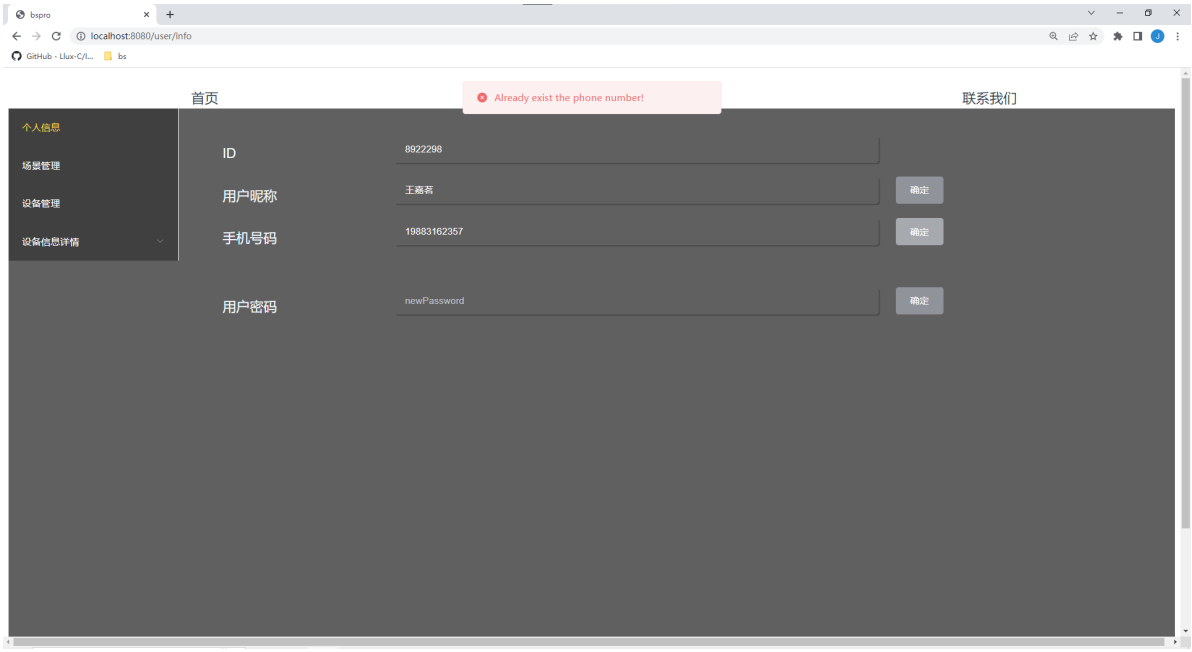
- 登陆后，点击左侧导航栏中的个人信息，跳转到个人信息界面。



- 修改密码，需要在用户密码栏中输入新的密码然后点击右边对应的确定，会弹出弹窗要求输入旧的密码，若输入正确则修改成功。

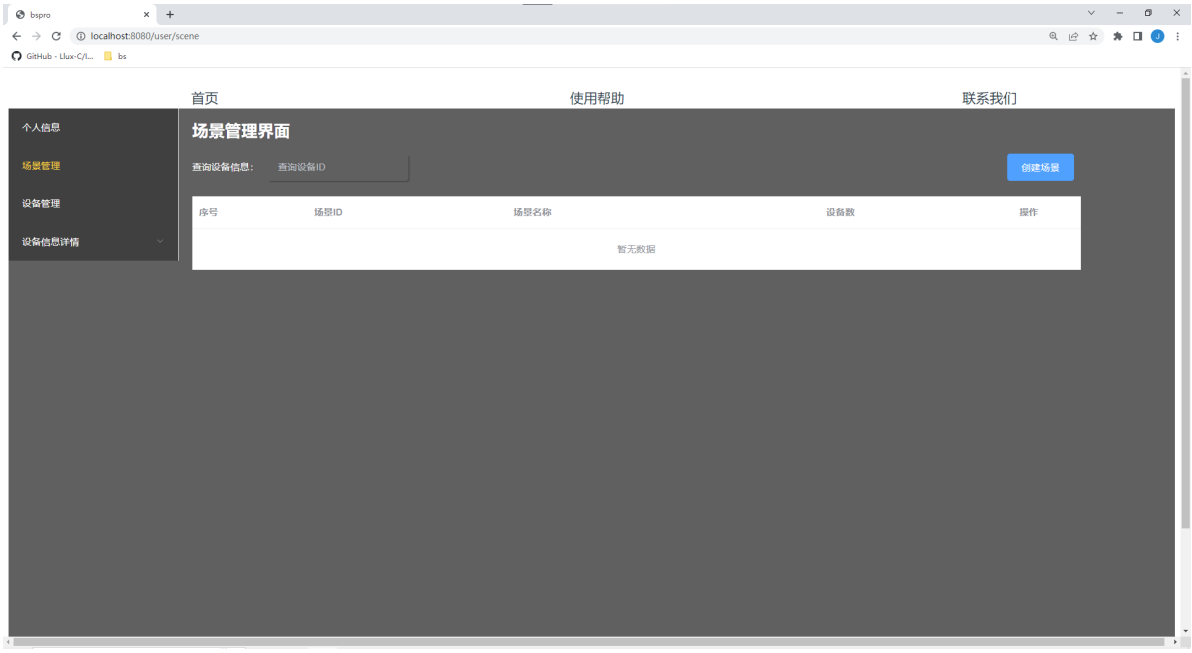


- 修改手机号码，不可与已注册过的用户使用相同的手机号码。若存在，则修改失败。

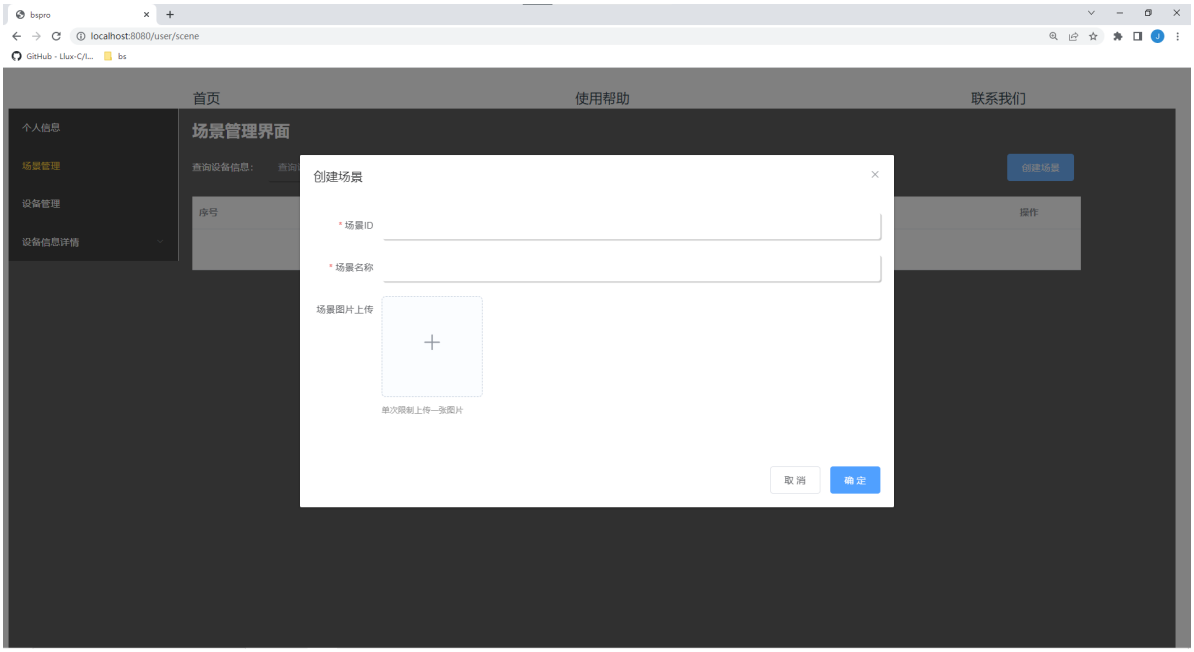


4.3 场景管理与创建

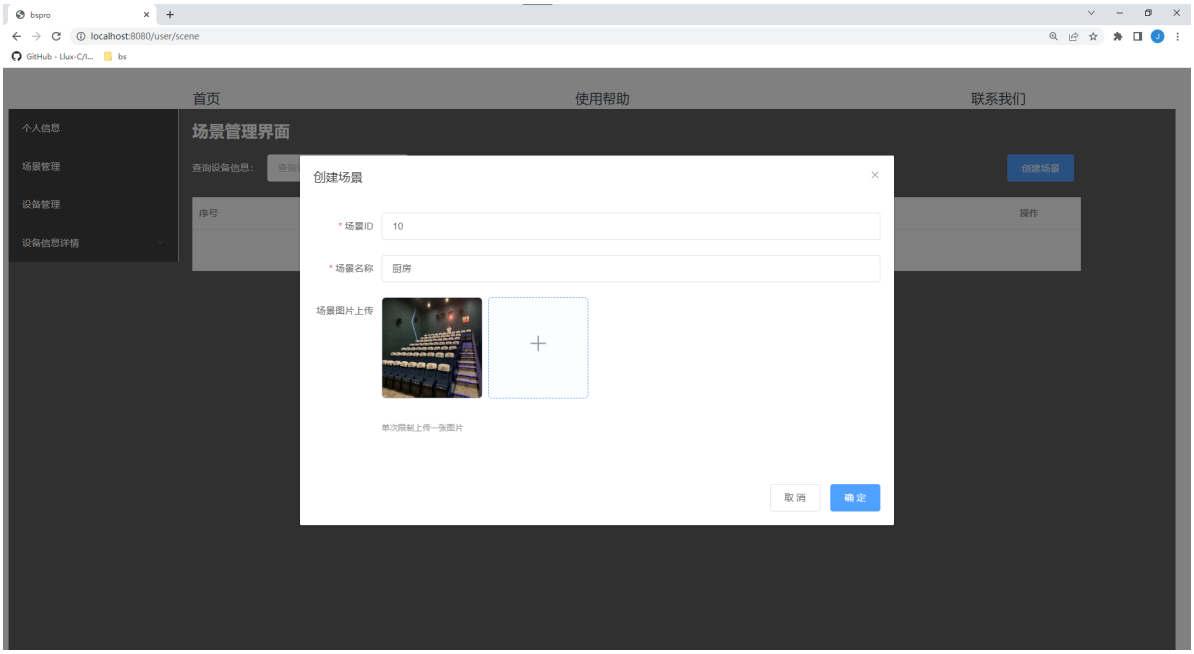
- 首先进入场景管理界面，如下所示为没有创建过任何场景的账号。



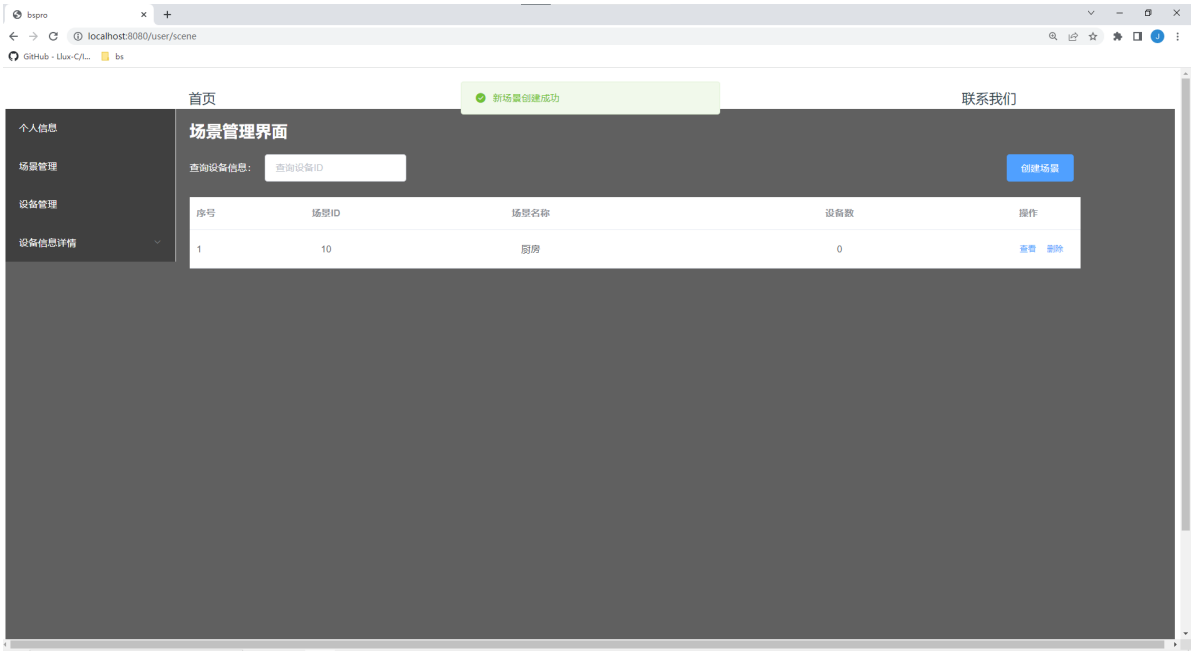
- 若需要创建场景，可点击右上角的蓝色按钮创建场景。



- 填入具体信息，传入场景照片后如下所示

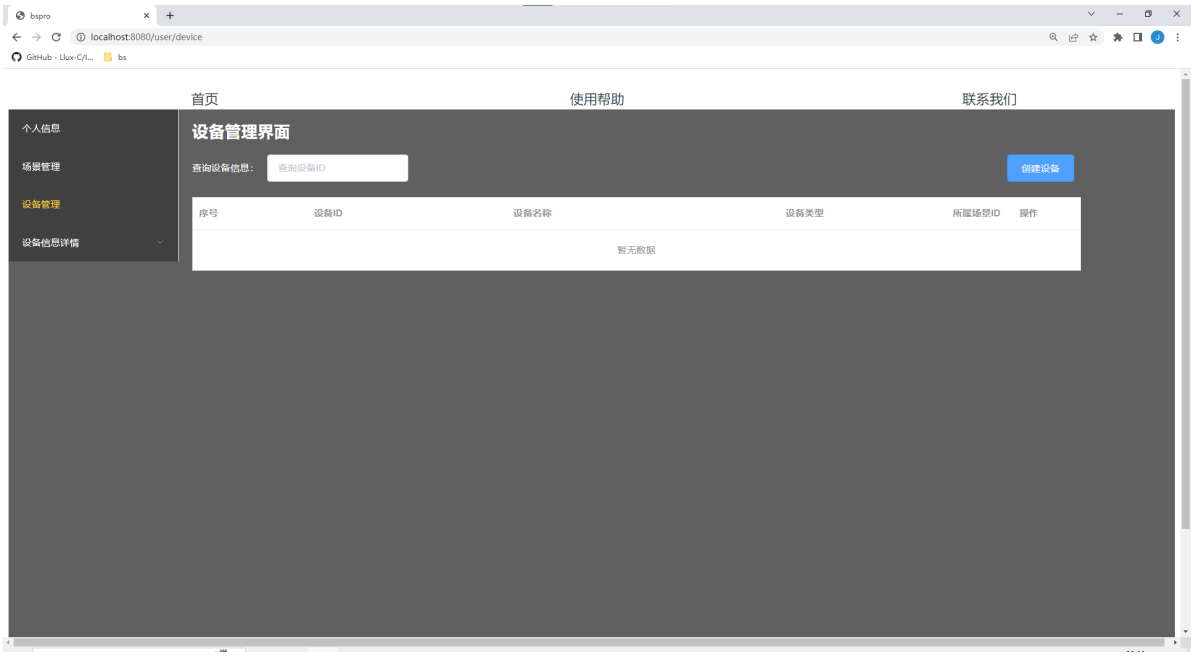


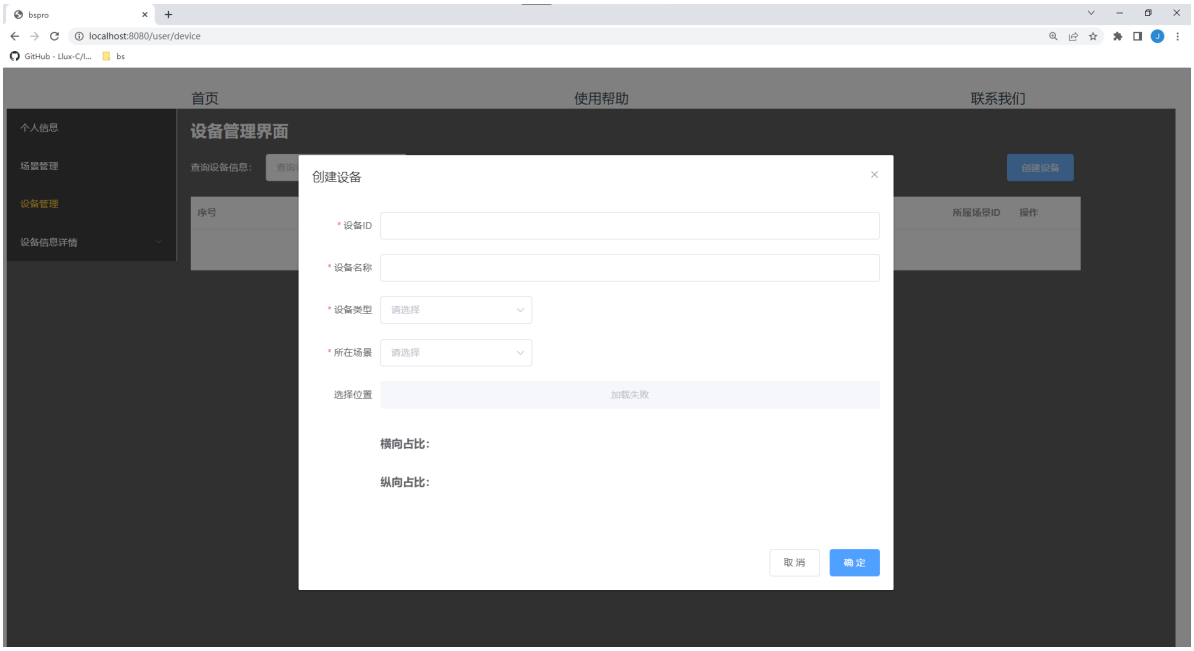
- 点击确定即创建成功，创建成功后，新创建的场景出现在场景管理界面。



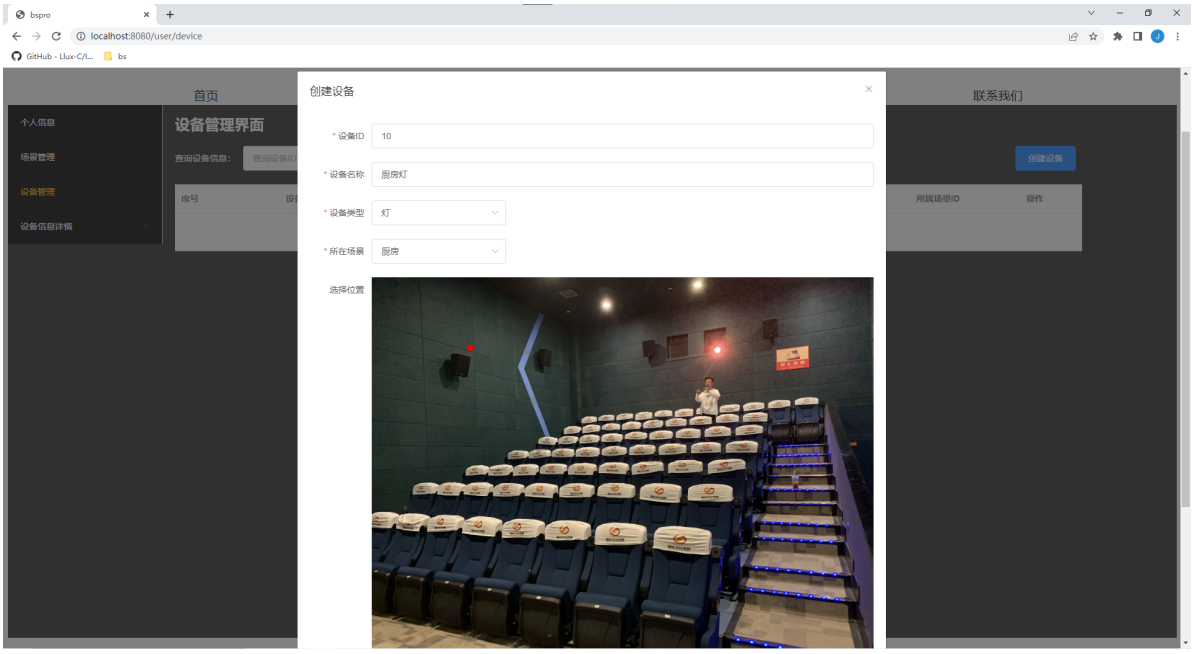
4.4 设备管理与创建

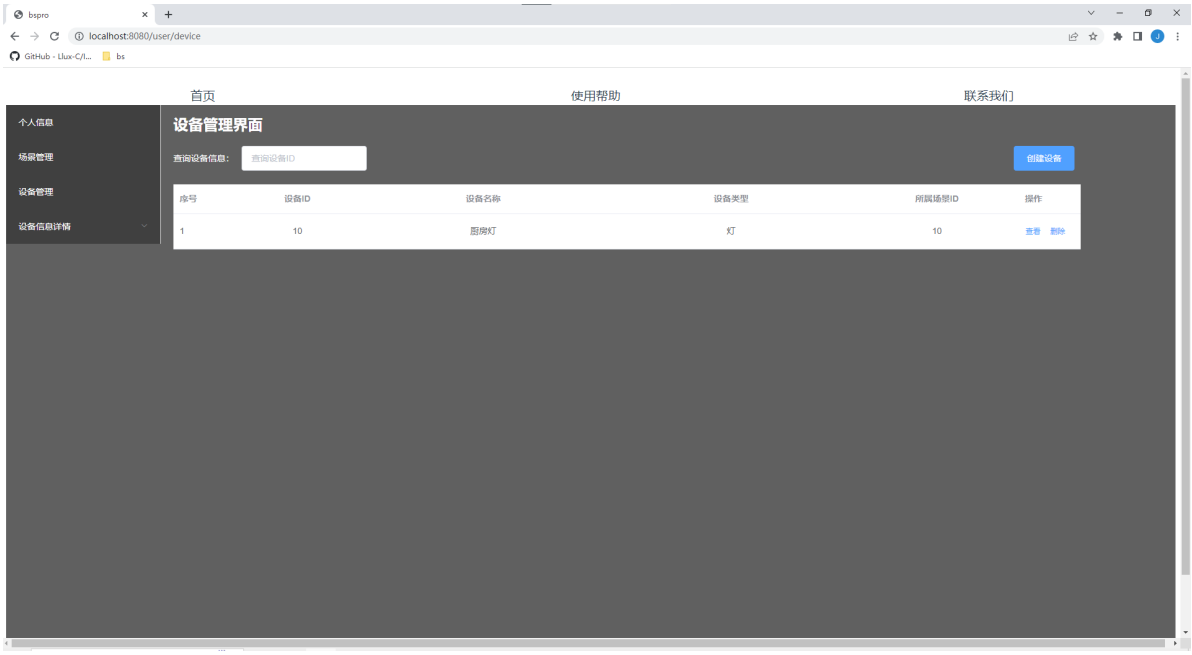
- 与场景创建管理步骤类似，下面展示各个步骤的截图，仅对不同之处做解释。



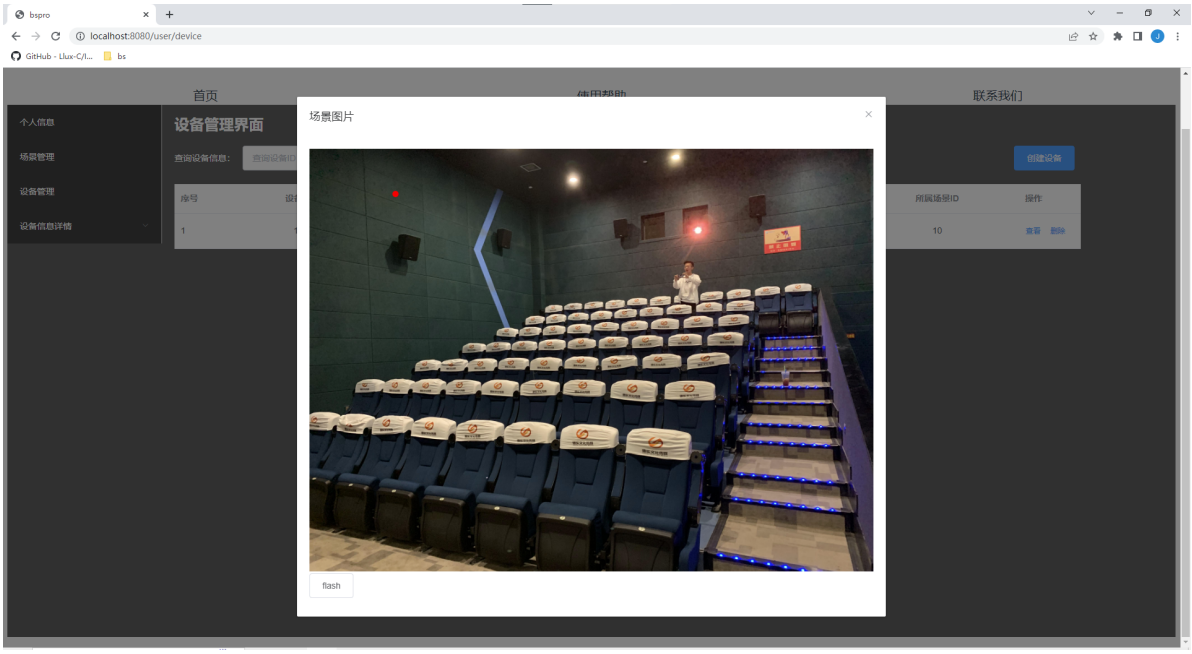


- 值得注意的是，当选择了所在场景之后，会出现场景的图片，然后点击图片即可确定设备摆放的位置。



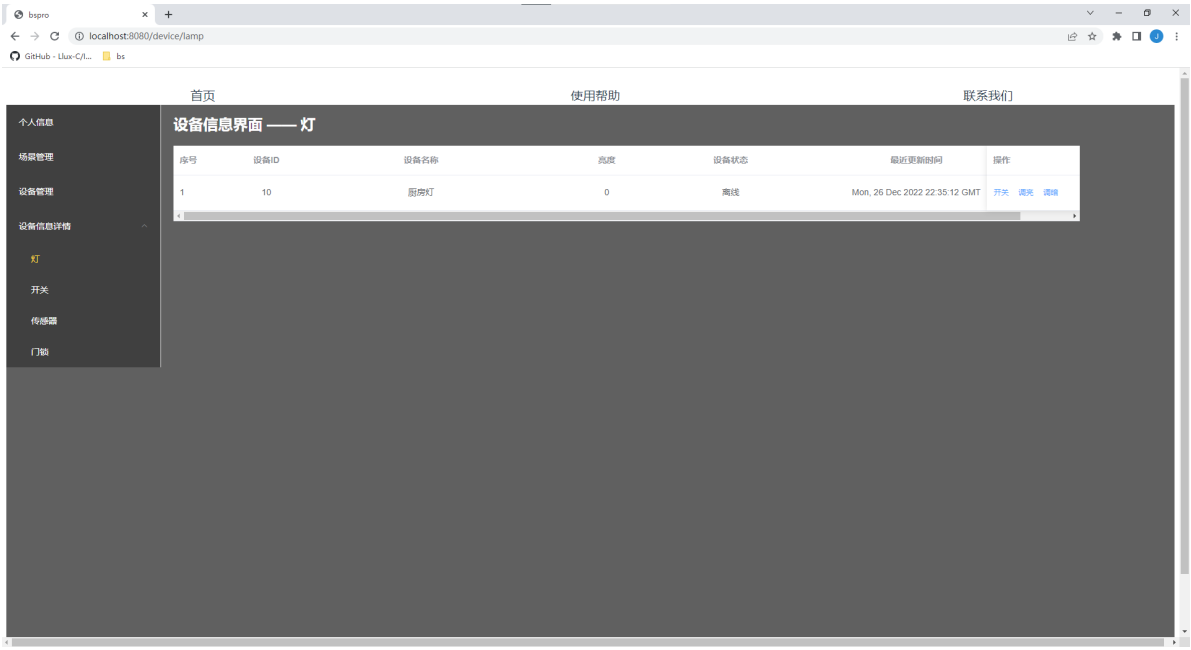


- 点击设备栏右边操作栏中的查看，即可查看设备的位置



4.4 设备具体上报信息查看

- 点击左侧导航栏的设备信息详情即可展开，作为演示，我们查看刚刚创建的灯



- 在此界面中可以实现对灯亮度以及开关和亮度的控制，可以查看设备的状态。可以看到此时设备状态为离线，当我们使用模拟终端发送信号之后，设备状态以及上报信息都会根据终端发送的信号有所变化。

```
send = ''
client.publish(MQTT_Topic, payload='10#2#1', qos=0) # light
```