

# 智能灯控前端

使用 TypeScript + Vue + Element Plus UI + axios 框架开发。有现代化的界面。响应式布局，可以适应多种设备。可以一键打开或关闭所有灯，可设置设备备注。可以显示设备是否在线等其他状态。



## 使用

当智能硬件连接服务器之后，界面上会自动添加设备。

当智能硬件断开服务器之后，界面上不会自动删除设备，而是显示离线状态。

要删除某个设备，请长按该设备，然后点击对话框的删除设备，删除完成后会弹出成功提示。

要为某个设备添加备注，请长按该设备，然后点击对话框的设备备注输入框，输入你想要设置的备注。

再点击保存。设置备注后设备名称会显示备注，而不再显示设备地址。

若要删除备注，只需要将设备备注改为空即可。

在设备离线时对开关的控制是无效的。

如果网络条件较差，可能会有一定的延迟。

## 部署

首先你需要安装 Docker 和 Docker-Compose

Docker 官方安装教程 <https://docs.docker.com/engine/install/>

通过脚本安装:

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
```

Docker-Compose 官方安装教程 <https://docs.docker.com/compose/install/>  
安装 Docker-Compose

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-  
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose  
sudo chmod +x /usr/local/bin/docker-compose
```

如果找不到 `docker-compose` 命令

```
sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

在树莓派等 Arm 平台

```
sudo apt install docker-compose
```

然后进入项目目录  
创建几个文件夹

```
mkdir ./log  
mkdir ./log/web  
mkdir ./build
```

开始构建

```
sudo docker-compose up -d
```

接下来会自动构建2个镜像 `smartlightfrontend_web` 和 `smartlightfrontend_npm-build`  
并创建2个服务 `smart-light-frontend` 和 `npm-build`

`npm-build` 是用来构建网站的，正常来说会在几分钟后停止运行

可以用 `sudo docker logs -f npm-build` 来查看构建日志

如果要重新构建，运行该服务即可 `sudo docker start npm-build`

完成后会监听 80 端口，如果要改为其它端口请修改 `docker-compose.yml` 文件

```
web:  
build:  
  context: ./nginx  
  dockerfile: Dockerfile  
container_name: acm-web-nginx  
hostname: acmweb  
volumes:  
  - ./build:/var/www/web  
  - ./nginx/conf.d:/etc/nginx/conf.d  
  - ./log:/var/log/nginx  
ports:  
  - "80:80"    #左边的 80 即为外部端口  
restart: always
```

## 一键部署脚本

```
sudo sh ./deploy.sh
```

## 更新

重新构建

```
sudo docker start npm-build
```

如果修改了 Nginx 的配置，还需要重启一下 `smartlightfrontend_web` 服务

```
sudo docker restart smartlightfrontend_web
```

## 一键更新脚本

```
sudo sh ./update.sh
```

## 开发

### 关于后端

后端数据会通过 Nginx 转发。

具体请参考 Nginx 配置文件 `nginx/conf.d/web.conf`

```
npm run dev
```

以开发模式运行项目。

可以在浏览器打开 <http://localhost:3000> 来查看效果。

页面会在你修改代码后这个更新，并在控制台显示错误。

```
npm run build
```

构建项目，输出到 `dist` 文件夹。

在构建是 vite 会自动优化，来得到更好的性能。

因为本项目使用的前端路由是基于http协议工作的，所以构建完成的文件是不能直接在浏览器打开查看。

如果要预览构建完成的效果，请使用 `serve dist` 命令。