

# 接口文档

---

## 接口文档

### 前后端API

#### user API

1. 注册用户
2. 检查账号密码

#### device API

1. 添加设备
2. 获取设备列表
3. 修改设备信息

4. 获取设备消息

#### message API

1. 获取设备位置列表
2. 获取某个设备的位置历史

#### mqtt API

1. 向某个设备的发布mqtt消息

### 后端与mqtt服务器 API

物联网设备向后端发布消息

后端向物联网设备发布消息

mqtt服务器发布流量消息

mqtt服务器发布设备连接/断连消息

## 前后端API

---

前后端使用http协议通讯

### user API

#### 1. 注册用户

##### 请求

- 方法: POST
- url: BASE\_URL/user/register
- 请求体:

```
{
  "username": String,
  "email": String,
  "password": String
}
```

## 响应

- 响应体:

```
{
  "status": 成功 -- "success"
           错误 -- "error"
  "message": 成功 -- "账号注册成功" ||
            错误 -- "用户名重复" ||
            -- "邮箱已注册账号"
}
```

## 2. 检查账号密码

### 请求

- 方法: POST
- url: BASE\_URL/user/check
- 请求体:

```
{
  "email": String,
  "password": String
}
```

### 响应

- 响应体:

成功:

```
{
  "status": "success",
  "username": String,
  "user_id": Number,
  "email": String
}
```

错误:

```
{
  "status": "error"
  "message": "密码错误" ||
            "邮箱未注册账号"
}
```

# device API

## 1. 添加设备

### 请求

- 方法: POST
- url: BASE\_URL/device/add
- 请求体:

```
{
  "user_id": Number,
  "device_id": String,
  "device_name": String,
  "device_location": String,
  "device_type": String
}
```

### 响应

- 响应体

成功:

```
{
  "status": "success",
  "message": "设备添加成功"
}
```

错误:

```
{
  "status": "error"
  "message": "密码错误" ||
             "邮箱未注册账号" ||
             "设备已经存在, 不可重复创建" ||
             "用户不存在"
}
```

## 2. 获取设备列表

### 请求

- 方法: GET
- url: BASE\_URL/device/get?user\_id={}

### 响应

- 响应体

成功:

```
{
  "status": "success",
  "devices": [
    {
      "id": String,
      "name": String,
      "location": String,
      "type": String,
      "status": String,
      "is_online": Boolean
    }
    ...
  ]
}
```

错误:

```
{
  "status": "error"
  "message": "当前用户不存在" ||
             str(e)
}
```

### 3. 修改设备信息

请求

- 方法: POST
- url: BASE\_URL/device/modify
- 请求体:

```
{
  "user_id": Number,
  "device_id": String,
  "device_name": String,
  "device_location": String,
  "device_type": String,
  "device_status": String,
  "is_online": Boolean
}
```

响应

- 成功:

```
{
  "status": "success",
  "message": "设备信息成功修改"
}
```

错误:

```
{
  "status": "error"
  "message": "设备不存在" ||
             "无效的JSON格式" ||
             str(e)
}
```

## 4. 获取设备消息

请求

- 方法: GET
- url: BASE\_URL/device/get\_message?user\_id={}&device\_id={}

响应

- 成功:

```
{
  "status": "success",
  "message_list": [
    {
      "type": String,
      "timestamp": DATETIME,
      "text": String
    }
  ]
}
```

错误:

```
{
  "status": "error"
  "message": "当前用户不存在" ||
             "当前设备不存在" ||
             str(e)
}
```

# message API

## 1. 获取设备位置列表

请求

- 方法: GET
- url: BASE\_URL/message/location\_list?user\_id={}

响应

- 成功:

```
{
  "status": "success",
  "position_list": [
    {
      "name": String,
      "id": String,
      "timestamp": DATETIME,
      "latitude": Number,
      "longitude": Number
    }
  ]
}
```

错误:

```
{
  "status": "error"
  "message": "请提供用户id" ||
             "当前用户不存在" ||
             str(e)
}
```

## 2. 获取某个设备的位置历史

请求

- 方法: GET
- url: BASE\_URL/message/location\_history?user\_id={}&device\_id={}

响应

- 成功:

```
{
  "status": "success",
  "history_list": [
    {
      "timestamp": DATETIME,
      "latitude": Number,
      "longitude": Number
    }
  ]
}
```

错误:

```
{
  "status": "error"
  "message": "当前用户不存在" ||
             "当前设备不存在" ||
             str(e)
}
```

## mqtt API

### 1. 向某个设备的发布mqtt消息

请求

- 方法: POST
- url: BASE\_URL/message/post
- 请求体:

```
{
  "device_id": String,
  "type": String,
  普通消息和告警消息:
  "text": String
  位置消息:
  "latitude": Number,
  "longitude": Number
}
```

响应

- 成功:

```
{
  "status": "success",
  "message": "接收到消息, message_id = {message.id}"
}
```

错误:

```
{
  "status": "error"
  "message": "消息类型不符合要求" ||
             "当前设备不存在" ||
             str(e)
}
```

## 后端与mqtt服务器 API

后端与mqtt服务器（本项目中使用emqx）使用mqtt协议通信

### 物联网设备向后端发布消息

topic: iot/{设备的id}

payload: JSON格式

- 普通消息和告警消息

```
{
  "type": String,
  "device_id": String,
  "text":String
}
```

- 位置消息

```
{
  "type": String,
  "device_id": String,
  "latitude": Number,
  "longitude": Number
}
```



## 后端向物联网设备发布消息

topic: response/ iot/{设备的id}

payload: 消息(String)

## mqtt服务器发布流量消息

使用EMQX服务器的系统消息

topic: \$SYS/brokers/+/metrics/bytes/received

payload: 这里的payload有很多信息，主要是用

## mqtt服务器发布设备连接/断连消息

这里将EMQX服务器将系统消息转发到http服务

- 方法: POST
- url: BASE\_URL/mqtt/online
- 请求体:

```
{
  这里的请求体由emqx服务器提供，有很多属性，只需要关注event属性，主要是判断是否是
  client.disconnected或者client.connected
  "event": String,
}
```

### 响应

- 成功:

```
{
  "status": "success",
  "message": "在线状态更新成功"
}
```

错误:

```
{
  "status": "error"
  "message": str(e)
}
```