

- 📖 一.用户认证模块 API 文档
  - 1.用户登录
    - 📄 请求参数
      - 示例请求体
    - ✅ 成功响应 (200 OK)
    - ❌ 失败响应 (401 Unauthorized)
    - ❌ 失败响应 (400 Bad Request )
    - 📌 认证说明
    - 登录说明
  - 2. 用户注册
    - 📄 请求参数
      - 示例请求体 (选择身份证号)
      - 示例请求体 (选择护照号)
      - 成功响应 (201 Created)
    - 422 验证错误 (400 Bad Request)
    - 手机号已存在 (409 Conflict)
    - 服务器内部错误 (500 Internal Server Error)
  - 📌 状态码说明
- 二。医生与患者关系管理系统 API 文档
  - 通用信息
  - 医生管理
    - 1. 获取医生列表
      - ✅ 响应
    - 失败响应
    - 2. 注册新医生
      - 📄 请求体
      - ✅ Response (成功)
    - 失败响应
  - 3.获取患者列表
    - ✅ 响应
    - 失败响应
  - 医生-患者关系管理
    - 4.获取医生-患者关系列表
      - ✅ 响应
    - 失败响应
    - 5.添加医生-患者关系
      - 📄 请求体

-  响应
- 失败响应
- 6.更新医生-患者关系
  -  请求体
  -  响应
- 失败响应
- 7.删除医生-患者关系
  -  请求体
  -  响应（成功）
- 失败响应
-  状态码说明
- 三. 患者管理系统 RESTful API 文档
  - 接口总览
  - 1. 获取某位医生的患者列表
    - 失败响应
  - 2. 获取某位患者的步态评估报告
    - 失败响应
- 四. CSV 数据上传接口文档
  - 接口总览
  - 1. 上传 IMU 数据 CSV 文件
  - 接口约束说明
  - 拦截器调试说明（前端开发用）
- 五.附加说明



## 一.用户认证模块 API 文档

本模块提供基础的用户登录与注册功能，采用 RESTful API 设计。



### 1.用户登录

- URL: `/api/session`
- 方法: `POST`
- 描述: 用户登录接口，根据手机号（姓名）和密码进行验证，成功后返回用户信息及权限角色。



## 请求参数

参数名	类型	是否必填	说明
username	string	是	手机号/姓名
password	string	是	密码

### 示例请求体

```
{
  "username": "doctor1",
  "password": "password"
}
```



## 成功响应（200 OK）

```
{
  "status": 200,
  "data": {
    "id": "101",
    "isAdmin": false,
    "isDoctor": true,
    "isPatient": false,
    "name": "医生1",
    "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJkb2N0b3IiwiiaWF0IjoxNjA3NzQ5MTYyL
CJleHBpcnkiOiJlMzQ5MTYyLzIwMjQCE-rH1QNGzmmx6Vmg6XYISbFQm9fg2ESyI9gyLQ",
  },
  "message": "登录成功"
}
```



## 失败响应（401 Unauthorized）

```
{
  "status": 401,
  "data": null,
  "message": "用户名或密码错误"
}
```

## ✖ 失败响应（400 Bad Request）

```
{
  "status": 400,
  "data": null,
  "message": "缺少 username 或 password 参数"
}
```

## 📌 认证说明

登录成功：成功登录后，返回的响应中会包含一个 **token** 字段，保存了经过加密的 JWT（JSON Web Token）。这个令牌是用来验证后续请求用户身份的。

认证过程：后续每个需要身份验证的 API 请求，都需要在请求头中附带 **Authorization** 字段，格式为：

```
Authorization: Bearer <token>
```

后端在接收到请求时，会解码并验证令牌，确保用户身份合法。如果令牌无效或已过期，将返回 **401 Unauthorized** 错误。

## 登录说明

医生和患者可通过手机号或姓名登录。管理员账号需预先设定，不提供 API 注册。

## ■ 2. 用户注册

- **URL:** `/api/users`
- **方法:** `POST`
- **描述:** 用户注册接口，提供姓名、电话、性别、出生日期和身份证号/护照信息，密码，成功后返回用户基本信息。



# 请求参数

参数名	类型	是否必填	说明
name	string	是	姓名
phone	string	是	电话号码
gender	string	是	性别（如 "男" 或 "女"）
birthDate	string	是	出生日期
idType	string	是	"身份证" 或 "护照"
idNumber	string	是	证件号码
password	string	是	密码

## 示例请求体（选择身份证号）

```
{
  "name": "新用户",
  "phone": "13812345678",
  "gender": "男",
  "birthDate": "1990-01-01",
  "idType": "身份证",
  "idNumber": "123456789012345678",
  "password": "123"
}
```

## 示例请求体（选择护照号）

```
{
  "name": "新用户",
  "phone": "13812345678",
  "gender": "男",
  "birthDate": "1990-01-01",
  "idType": "护照",
  "idNumber": "p12345678901",
  "password": "123"
}
```

## 成功响应（201 Created）

```
{
  "status": 201,
  "data": {
    "id": "202",
    "name": "新用户",
    "phone": "13812345678",
    "gender": "男",
    "birthDate": "1990-01-01",
    "idType": "身份证",
    "idNumber": "123456789012345678"
  },
  "message": "注册成功"
}
```

## 422 验证错误 (400 Bad Request)

```
{
  "status": 400,
  "data": null,
  "message": "请求参数错误: 电话号码格式不正确" // 示例消息
}
```

## 手机号已存在 (409 Conflict)

```
{
  "status": 409,
  "data": null,
  "message": "手机号已被注册",
}
```

## 服务器内部错误 (500 Internal Server Error)

```
{
  "status": 500,
  "data": null,
  "message": "服务器内部错误, 请联系管理员"
}
```

---



# 状态码说明

状态码	含义
200	请求成功
201	创建成功
401	未授权（用户名或密码错误）
409	冲突（如用户名已存在）
500	服务器内部错误

## 二。医生与患者关系管理系统 API 文档

本 API 文档基于 RESTful 规范，支持医生注册、获取医生/患者信息、以及医生-患者关系的增删改查。所有接口需要管理员权限

### 通用信息

- 请求地址统一前缀：`/api/admin`
- 请求头：

`Content-Type: application/json Authorization: Bearer <token>`



## 医生管理

### 1. 获取医生列表

- URL：`/api/admin/doctors`
- 方法：GET
- 描述：获取所有已注册的医生信息

✓ 响应

```
{
  "status": 200,
  "data": [
    {
      "id": "1",
      "name": "张三",
      "phone": "13800001111",
      "hospital": "北京协和医院",
      "department": "内科"
    },
    {
      "id": "2",
      "name": "李四",
      "phone": "13800002222",
      "hospital": "上海瑞金医院",
      "department": "外科"
    }
  ],
  "message": "获取医生列表成功"
}
```

## 失败响应

// 401 Unauthorized (未提供令牌或令牌无效)

```
{ "status": 401, "data": null, "message": "未授权访问" }
```

// 403 Forbidden (用户非管理员)

```
{ "status": 403, "data": null, "message": "权限不足" }
```

// 500 Internal Server Error

```
{ "status": 500, "data": null, "message": "服务器内部错误" }
```

---

### • 字段说明:

字段	类型	描述
status	number	状态码（如 200）



字段	类型	描述
data	array	医生信息数组

data 数组内对象字段说明：

字段	类型	描述
id	string	医生唯一 ID
name	string	姓名
phone	string	电话号码
hospital	string	所在医院
department	string	科室

## 2. 注册新医生

- 接口路径: /api/admin/doctors
- 请求方法: POST
- 接口描述: 通过提供密码、姓名、手机号、医院和科室信息来注册一个新的医生账号。

### 请求体

```
{
  "password": "123456",
  "name": "张三",
  "phone": "13812345678",
  "hospital": "Beijing General Hospital",
  "department": "Cardiology"
}
```

### Response (成功)

```
{
  "status": 201,
  "data": {
    "id": "3",
```

```
{
  "name": "张三",
  "phone": "13812345678",
  "hospital": "Beijing General Hospital",
  "department": "Cardiology"
},
{
  "message": "医生注册成功"
}
```

## 失败响应

// 401 Unauthorized (未提供令牌或令牌无效)

```
{ "status": 401, "data": null, "message": "未授权访问" }
```

// 403 Forbidden (用户非管理员)

```
{ "status": 403, "data": null, "message": "权限不足" }
```

// 500 Internal Server Error

```
{ "status": 500, "data": null, "message": "服务器内部错误" }
```

// 409 Conflict (手机号或姓名已存在)

```
{ "status": 409, "data": null, "message": "手机号或姓名已被注册" }
```

// 400 Bad Request (参数校验失败)

```
{ "status": 400, "data": null, "message": "请求参数错误: 缺少字段" }
```

- 字段说明:** | 字段 | 类型 | 描述 | |-----|-----|-----| | **id** | string | 医生唯一 ID | | **name** | string | 医生姓名 | | **phone** | string | 电话号码 | | **hospital** | string | 医院名称 | | **department** | string | 科室名称 |



## 3. 获取患者列表

- **URL:** `/api/admin/patients`
- **方法:** GET
- **描述:** 获取所有患者信息

### ✓ 响应

```
{
  "status": 200,
  "data": [
    {
      "id": "1",
      "name": "王五"
    },
    {
      "id": "2",
      "name": "赵六"
    }
  ],
  "message": "获取患者列表成功"
}
```

若无患者data 返回空数组[]

## 失败响应

// 401 Unauthorized (未提供令牌或令牌无效)

```
{ "status": 401, "data": null, "message": "未授权访问" }
```

// 403 Forbidden (用户非管理员)

```
{ "status": 403, "data": null, "message": "权限不足" }
```

// 500 Internal Server Error

```
{ "status": 500, "data": null, "message": "服务器内部错误" }
```

## 医生-患者关系管理

### 4. 获取医生-患者关系列表

- URL: `/api/admin/relations`
- 方法: GET
- 描述: 返回所有医生与患者之间的关联信息

#### ✓ 响应

```
{
  "status": 200,
  "data": [
    {
      "doctorId": "1",
      "doctorName": "张三",
      "patientId": "2",
      "patientName": "赵六"
    }
  ],
  "message": "获取医生-患者关系列表成功"
}
```

### 失败响应

// 401 Unauthorized (未提供令牌或令牌无效)

```
{ "status": 401, "data": null, "message": "未授权访问" }
```

// 403 Forbidden (用户非管理员)

```
{ "status": 403, "data": null, "message": "权限不足" }
```

// 500 Internal Server Error

```
{ "status": 500, "data": null, "message": "服务器内部错误" }
```

## 5. 添加医生-患者关系

- URL: `/api/admin/relations`
- 方法: POST
- 描述: 添加医生与患者之间的关系

### 请求体

```
{  
  "doctorId": "1",  
  "patientId": "2"  
}
```

### 响应

```
{  
  "status": 201,  
  "data": {  
    "doctorId": "1",  
    "patientId": "2"  
  },  
  "message": "医生-患者关系添加成功"  
}
```

## 失败响应

// 401 Unauthorized (未提供令牌或令牌无效)

```
{ "status": 401, "data": null, "message": "未授权访问" }
```

// 403 Forbidden (用户非管理员)

```
{ "status": 403, "data": null, "message": "权限不足" }
```

// 500 Internal Server Error

```
{ "status": 500, "data": null, "message": "服务器内部错误" }
```

// 409 Conflict (关系已存在)

```
{ "status": 409, "data": null, "message": "该医患关系已存在" }
```

// 400 Bad Request (参数错误或 Doctor/Patient ID 不存在)

```
{ "status": 400, "data": null, "message": "请求参数错误: 医生 ID '1' 不存在" }
```

---

## 6.更新医生-患者关系

- URL: `/api/admin/relations`
- 方法: PUT
- 描述: 更新已有的医生与患者的关系

### 请求体

```
{  
  "oldDoctorId": "1",  
  "oldPatientId": "2",  
  "doctorId": "3",  
  "patientId": "4"  
}
```

### 响应

```
{  
  "status": 200,  
}
```

```
"data": {
  "doctorId": "3",
  "patientId": "4"
},
"message": "更新医生-患者关系成功"
}
```

## 失败响应

// 401 Unauthorized (未提供令牌或令牌无效)

```
{ "status": 401, "data": null, "message": "未授权访问" }
```

// 403 Forbidden (用户非管理员)

```
{ "status": 403, "data": null, "message": "权限不足" }
```

// 500 Internal Server Error

```
{ "status": 500, "data": null, "message": "服务器内部错误" }
```

// 409 Conflict (关系已存在)

```
{ "status": 409, "data": null, "message": "该医患关系已存在" }
```

// 400 Bad Request (参数错误或 Doctor/Patient ID 不存在)

```
{ "status": 400, "data": null, "message": "请求参数错误: 医生 ID '1' 不存在" }
```

---

## 7.删除医生-患者关系

- **URL:** `/api/admin/relations`
- **描述:** 根据关系 ID 删除对应的医生-患者关系

- 方法: DELETE

-  请求体

```
{
  "doctorId": "1",
  "patientId": "2"
}
```

## ✅ 响应 (成功)

```
{
  "status": 204,
  "data": null,
  "message": "删除医生-患者关系成功"
}
```

# 失败响应

// 401 Unauthorized (未提供令牌或令牌无效)

```
{ "status": 401, "data": null, "message": "未授权访问" }
```

// 403 Forbidden (用户非管理员)

```
{ "status": 403, "data": null, "message": "权限不足" }
```

// 500 Internal Server Error

```
{ "status": 500, "data": null, "message": "服务器内部错误" }
```

// 404 Not Found (指定的 doctorId/patientId 关系不存在)

```
{ "status": 404, "data": null, "message": "未找到要删除的医患关系" }
```



// 400 Bad Request (参数错误或 Doctor/Patient ID 不存在)

```
{ "status": 400, "data": null, "message": "请求参数错误" }
```

## 状态码说明

状态码	含义
200	请求成功
201	创建成功
204	删除成功，无内容返回
400	错误的请求参数
401	未授权访问
403	禁止访问 (Forbidden) - 权限不足
404	请求资源不存在
409	数据冲突（如用户名重复）
500	服务器错误

## 三. 患者管理系统 RESTful API 文档

该文档描述了患者管理系统中使用的 RESTful API 接口，包括医生关联患者的获取接口和患者报告数据的获取接口。通用信息 认证要求： 本模块所有接口均需要有效的 Authorization: Bearer 请求头。具体权限要求见各接口说明。响应结构： 遵循统一结构 { status, data, message }。

## 接口总览

方法	接口路径	描述
GET	/api/doctor/:doctorId/patients	获取某位医生的患者列表
GET	/api/patient/:patientId/reports	获取指定患者的步态报告列表

## 1. 获取某位医生的患者列表

- 接口路径： /api/doctor/:doctorId/patients
- 请求方法： GET
- 请求参数：
  - doctorId (路径参数)：医生的唯一标识（如：101）(类型：string)
- 请求示例：

```
GET /api/doctor/101/patients
```

- 响应示例 (200 OK)：

```
{
  "status": 200,
  "data": [
    {
      "id": "1001",
      "name": "张三",
      "phone": "13800001111",
      "gender": "男",
      "birthDate": "1980-05-20",
      "idNumber": "110101198005201234"
    },
    {
      "id": "1002",
      "name": "李四",
      "phone": "13800002222",
      "gender": "女",
      "birthDate": "1992-08-15",
      "idNumber": "110101199208152345"
    },
    {
      "id": "1003",
      "name": "王五",
      "phone": "13800003333",
```

```
    "gender": "男",
    "birthDate": "1965-12-01",
    "idNumber": "护照A12345678"
  }
]
"message": "success",
}
```

• 字段说明:

字段	类型	描述
status	number	状态码 (如 200)
message	string	状态信息
data	array	患者信息数组

data 数组内对象字段说明:

字段	类型	描述
id	string	患者唯一 ID
name	string	姓名
phone	string	电话号码
gender	string	性别 (男/女)
birthDate	string	出生日期 (格式: YYYY-MM-DD)
idNumber	string	身份证号或护照号码

## 失败响应

// 401 Unauthorized (未提供令牌或令牌无效)

```
{ "status": 401, "data": null, "message": "未授权访问" }
```

// 403 Forbidden (登录用户不是该医生且非管理员)

```
{ "status": 403, "data": null, "message": "权限不足, 无法查看该医生的患者列表" }
```

// 500 Internal Server Error

```
{ "status": 500, "data": null, "message": "服务器内部错误" }
```

// 404 Conflict (医生 ID 不存在)

```
{ "status": 404, "data": null, "message": "未找到指定的医生" }
```

## 2. 获取某位患者的步态评估报告

- 接口路径： `/api/patient/:patientId/reports`
- 请求方法： `GET`
- 请求参数：
  - `patientId` (路径参数)：患者的唯一标识（如：1001）(类型：string)
- 请求示例：

```
GET /api/patient/1001/reports
```

- 响应示例 (200 OK)：

```
{
  "status": 200,
  "data": [
    {
      "date": "2025-04-22",
      "type": "最新评估",
      "summary": "最新步态评估，各项指标稳定。",
      "reportData": {
        "运动幅度": [1,2,3,4,5,6,7,8,9,10,11,12],
        "得分": [1,2,3,4,5,6,7,8,9,10,11,12]
      }
    },
    {
      "date": "2025-04-18",
```

```
"type": "中期复查",
"summary": "中期复查，左腿活动度有所改善。",
"reportData": {
  "运动幅度": [10,9,1,2,3,4,5,6,7,8,9,10],
  "得分": [100,100,1,0,100,1,2,3,5,9,13,66]
}
},
"message": "查询成功",
}
//如果该患者无报告，reportData 为空数组 []
```

• 字段说明：

字段	类型	描述
status	number	状态码（如 200）
message	string	状态信息
data	array	患者信息数组

data 数组内对象字段说明：

字段	类型	描述
date	string	报告日期（YYYY-MM-DD）
type	string	报告类型（如初步评估）
summary	string	报告摘要描述
reportData	object	报告数据（包含指标数组）

- `运动幅度`：`number[]`，12项指标
- `得分`：`number[]`，12项分数

- 排序说明：
- 接口默认按 date 字段降序排序（最新在前）

# 失败响应

// 401 Unauthorized (未提供令牌或令牌无效)

```
{ "status": 401, "data": null, "message": "未授权访问" }
```

// 403 Forbidden (登录用户非该患者关联医生且非管理员)

```
{ "status": 403, "data": null, "message": "权限不足，无法查看该患者的报告" }
```

// 500 Internal Server Error

```
{ "status": 500, "data": null, "message": "服务器内部错误" }
```

// 404 Conflict (患者 ID 不存在)

```
{ "status": 404, "data": null, "message": "未找到指定的患者" }
```

---

---

## 四. CSV 数据上传接口文档

本节文档描述了患者管理系统中用于上传 IMU 数据 CSV 文件的 RESTful API 接口，遵循 RESTful API 设计规范。

---

### 接口总览

方法	接口路径	描述
POST	/api/upload/csv	上传 IMU 数据 CSV 文件

---

### 1. 上传 IMU 数据 CSV 文件

- 接口路径： /api/upload/csv
- 请求方法： POST

- 请求类型： `multipart/form-data`
- 描述： 向后端上传采集到的 IMU 数据 CSV 文件，同时附带患者 ID 信息。后端将保存并解析数据。
- 请求参数：

参数名	类型	是否必须	描述
file	File	是	CSV 文件内容（IMU数据）
patientId	string	是	患者的唯一标识

- 请求示例：

```
POST /api/upload/csv HTTP/1.1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary

-----WebKitFormBoundary
Content-Disposition: form-data; name="file"; filename="imu_data_2025-04-24.csv"
Content-Type: text/csv

时间戳,设备ID,设备名称,AccX(g),AccY(g),AccZ(g),GyroX(°/s),GyroY(°/s),GyroZ(°/s),Roll(°),Pitch(°),Yaw(°)
"2025-04-24T10:00:00",001,"IMU_A",0.01,0.02,0.98,0.1,0.1,0.2,10.0,5.0,0.0

-----WebKitFormBoundary
Content-Disposition: form-data; name="patientId"

1001
-----WebKitFormBoundary--
```

- 响应示例（200 OK）：

```
{
  "status": 200,
  "data": {
    "receivedAt": "2025-04-24T10:30:00.123Z"
  },
  "message": "CSV 文件上传成功",
}
```

- 响应字段说明：

字段名	类型	描述
status	number	状态码，200 表示成功
message	string	响应信息
data	object	返回的附加数据
receivedAt	string	后端接收到请求的时间戳（ISO格式）

• 错误响应示例（400 Bad Request）：

// 400 Bad Request (缺少参数、文件格式错误、文件过大、CSV内容不符合要求、Patient ID不存在等)

```
{ "status": 400, "data": null, "message": "请求错误: 缺少必需的 patientId 参数" }
{ "status": 400, "data": null, "message": "请求错误: 文件类型必须为 CSV" }
{ "status": 400, "data": null, "message": "请求错误: CSV 文件缺少必需的 '时间戳' 列" }
```

// 401 Unauthorized (未登录或令牌无效)

```
{ "status": 401, "data": null, "message": "未授权访问" }
```

// 403 Forbidden (无权为该患者上传数据)

```
{ "status": 403, "data": null, "message": "权限不足, 无法为该患者上传数据" }
```

// 500 Internal Server Error (服务器处理文件出错)

```
{ "status": 500, "data": null, "message": "服务器处理 CSV 文件时出错" }
```

# 接口约束说明

- 文件必须为 .csv 格式。
- 必须提供 patientId，否则上传失败。
- 上传内容必须包含至少一行数据（不包括表头）。



- Patient ID 校验： 后端应校验 patientId 是否有效存在。

---

## 拦截器调试说明（前端开发用）

---

当前系统实现了 `window.fetch` 的拦截器用于前端模拟测试：

- 拦截 `POST /api/upload/csv` 请求，并返回模拟的 200 成功响应。
- 模拟响应数据格式如下：

```
{
  "status": 200,
  "data": {
    "receivedAt": "2025-04-24T10:00:00.000Z"
  },
  "message": "模拟：CSV 文件上传成功 (Fetch Interceptor)",
}
```

用于前端调试时无须后端真实服务，可直接验证上传逻辑和用户提示。

## 五.附加说明

---

- 所有接口返回的时间格式统一为 `YYYY-MM-DD`
- 所有数值均为模拟数据，仅供测试使用
- API 支持通过 `Mock.js` 进行本地模拟测试
- 若对某个接口响应结构存在疑问，建议使用调试工具如 Postman、curl 测试