

# 智能交通管理系统 - 接口与功能说明文档

## 一、项目概述

这是一个智能交通管理系统后端，主要功能包括：

- 用户认证与权限管理（管理员/交警）
- 违章记录管理（上报、查询、处理）
- 交通信号灯控制（多方向、多转弯类型）
- 实时交通监控（路口状态、热力图、统计数据）
- AI违章检测集成（与大模型算法对接）

## 二、所有可用接口汇总

基础URL： `http://localhost:8080/api`

### 1. 认证模块 `/api/auth`

接口	方法	说明	是否需要认证
<code>/api/auth/login</code>	POST	用户登录	✗
<code>/api/auth/logout</code>	POST	用户登出	✓
<code>/api/auth/me</code>	GET	获取当前用户信息	✓

### 2. 管理员模块 `/api/admin`

接口	方法	说明	权限要求
<code>/api/admin/police</code>	POST	创建交警账号	ADMIN
<code>/api/admin/police</code>	GET	获取交警列表（分页）	ADMIN
<code>/api/admin/police/{id}/status</code>	PUT	更新账号状态	ADMIN

### 3. 违章管理模块 `/api/violations`

接口	方法	说明	是否需要认证
<code>/api/violations/report</code>	POST	上报违章行为	✗ (给AI调用)
<code>/api/violations</code>	GET	查询违章记录列表	✗
<code>/api/violations/{id}</code>	GET	查看单条违章详情	✗

接口	方法	说明	是否需要认证
/api/violations/{id}/process	PUT	处理违章	✗
/api/violations/increment	POST	增加违章计数	✗
/api/violations/count	GET	获取违章总数	✗

## 4. 信号灯控制模块 /api/signals

接口	方法	说明	权限要求
/api/signals	GET	获取所有路口信号灯配置	ADMIN/POLICE
/api/signals/{id}/config	GET	获取指定信号灯配置	ADMIN/POLICE
/api/signals/{id}/adjust	POST	调整信号灯配置	ADMIN/POLICE

## 5. 多方向交通控制模块 /api/multi-direction-traffic

接口	方法	说明
/api/multi-direction-traffic/intersections/{intersectionId}/directions	GET	获取路口所有方向配置
/api/multi-direction-traffic/intersections/{intersectionId}/status	GET	获取路口所有方向信号灯状态
/api/multi-direction-traffic/intersections/{intersectionId}/directions/{direction}/turns/{turnType}/status	GET	获取指定方向指定转弯类型信号灯状态
/api/multi-direction-traffic/intersections/{intersectionId}/directions/{direction}/config	GET	获取指定方向配置详情
/api/multi-direction-traffic/intersections/{intersectionId}/simulate	POST	模拟设置信号灯状态
/api/multi-direction-traffic/intersections/{intersectionId}/validate-violation	POST	验证违章是否成立
/api/multi-direction-traffic/intersections/{intersectionId}/cache	DELETE	清除路口缓存
/api/multi-direction-traffic/metadata	GET	获取支持的方向和类型元数据

## 6. 交通监控模块 /api/intersections & /api/dashboard

接口	方法	说明
/api/intersections	GET	获取所有路口列表及基础信息
/api/intersections/{id}/realtime	GET	获取某路口实时数据
/api/intersections/{id}	POST	设置路口数据 (测试用)
/api/dashboard/stats	GET	获取宏观交通趋势统计
/api/dashboard/stats	POST	设置统计数据 (测试用)
/api/dashboard/heatmap	GET	获取车辆分布热力图数据
/api/dashboard/heatmap	POST	设置热力图数据 (测试用)

## 7. 违章检测模块 /api/violation-detection (AI对接核心接口)

接口	方法	说明
/api/violation-detection/detect-frame	POST	AI上报检测帧 (核心)
/api/violation-detection/directions/{direction}/detect-frame	POST	方向特定的违章检测
/api/violation-detection/ai-callback	POST	AI回调接口
/api/violation-detection/ai-completed	POST	AI分析完成通知
/api/violation-detection/tasks/statistics	GET	获取任务统计
/api/violation-detection/intersection/{intersectionId}/tasks	GET	获取路口任务列表
/api/violation-detection/task/{taskId}/status	GET	获取任务状态

## 8. 系统健康检查 /api

接口	方法	说明
/api/health	GET	健康检查
/api/	GET	根路径欢迎信息
/api/ai-integration/status	GET	AI服务状态
/api/ai-integration/health	GET	AI健康检查

## 三、给路的对接文档（前端）

### 统一响应格式

所有接口返回统一的 `ApiResponse<T>` 格式：

```
1  {
2      "code": 200,
3      "message": "success",
4      "data": { ... },
5      "timestamp": "2025-12-22T10:30:00"
6 }
```

错误响应：

```
1  {
2      "code": 401,
3      "message": "未授权访问",
4      "data": null,
5      "timestamp": "2025-12-22T10:30:00"
6 }
```

### 1. 用户登录

`POST /api/auth/login`

请求体：

```
1  {
2      "username": "admin",
3      "password": "123456"
4 }
```

响应体：

```
1  {
2      "code": 200,
3      "message": "success",
4      "data": {
5          "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
6          "tokenType": "bearer",
7          "userInfo": {
8              "id": 1,
9              "role": "ADMIN",
10             "fullName": "系统管理员",
11             "username": "admin"
12         }
13     },
14     "timestamp": "2025-12-22T10:30:00"
15 }
```

后续请求头：

```
1 | Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

## 2. 创建交警账号

**POST** /api/admin/police

**请求头:** 需要Bearer Token (管理员权限)

**请求体:**

```
1 | {
2 |     "username": "police001",
3 |     "password": "123456",
4 |     "fullName": "张警官",
5 |     "policeNumber": "P001"
6 | }
```

**响应体:**

```
1 | {
2 |     "code": 200,
3 |     "message": "创建成功",
4 |     "data": {
5 |         "id": 2,
6 |         "username": "police001",
7 |         "fullName": "张警官",
8 |         "role": "POLICE",
9 |         "policeNumber": "P001",
10 |        "status": "ACTIVE",
11 |        "createdAt": "2025-12-22T10:30:00",
12 |        "updatedAt": "2025-12-22T10:30:00"
13 |     }
14 | }
```

## 3. 获取交警列表 (分页)

**GET** /api/admin/police?page=1&size=10

**响应体:**

```
1 | {
2 |     "code": 200,
3 |     "message": "success",
4 |     "data": {
5 |         "content": [
6 |             {
7 |                 "id": 2,
8 |                 "username": "police001",
9 |                 "fullName": "张警官",
10 |                "role": "POLICE",
11 |                "policeNumber": "P001",
12 |                "status": "ACTIVE"
```

```
13     }
14   ],
15   "page": 1,
16   "size": 10,
17   "total": 1,
18   "totalPages": 1
19 }
20 }
```

## 4. 更新账号状态

**PUT** /api/admin/police/{id}/status

请求体:

```
1 {
2   "status": "SUSPENDED"
3 }
```

可选值: ACTIVE (启用) / SUSPENDED (禁用)

## 5. 查询违章记录列表

**GET** /api/violations?page=1&size=10

响应体:

```
1 [
2   {
3     "id": 1,
4     "intersectionId": 1,
5     "direction": "SOUTH",
6     "turnType": "STRAIGHT",
7     "plateNumber": "京A12345",
8     "violationType": "RED_LIGHT",
9     "imageUrl": "https://example.com/image.jpg",
10    "aiConfidence": 0.95,
11    "occurredAt": "2025-12-22T10:30:00",
12    "status": "PENDING",
13    "appealStatus": "NO_APPEAL"
14  }
15 ]
```

违章类型枚举 (violationType) :

- RED\_LIGHT - 闯红灯
- WRONG WAY - 逆行
- CROSS\_SOLID\_LINE - 跨实线
- ILLEGAL\_TURN - 违法转弯
- SPEEDING - 超速

- PARKING\_VIOLATION - 违章停车
- OTHER - 其他

#### 处理状态枚举 (status) :

- PENDING - 待处理
- CONFIRMED - 已确认
- REJECTED - 已驳回

---

## 6. 查看违章详情

GET /api/violations/{id}

响应体:

```
1  {
2    "id": 1,
3    "intersectionId": 1,
4    "direction": "SOUTH",
5    "turnType": "STRAIGHT",
6    "plateNumber": "京A12345",
7    "violationType": "RED_LIGHT",
8    "imageUrl": "https://example.com/image.jpg",
9    "aiConfidence": 0.95,
10   "occurredAt": "2025-12-22T10:30:00",
11   "status": "PENDING",
12   "processedBy": null,
13   "processedAt": null,
14   "penaltyAmount": null,
15   "reviewNotes": null,
16   "appealStatus": "NO_APPEAL"
17 }
```

---

## 7. 处理违章

PUT /api/violations/{id}/process

请求体:

```
1  {
2    "status": "CONFIRMED",
3    "penaltyAmount": 200.00,
4    "reviewNotes": "确认闯红灯违章"
5 }
```

## 8. 获取所有路口信号灯配置

**GET** /api/signals

响应体:

```
1  {
2      "code": 200,
3      "data": [
4          {
5              "intersectionId": 1,
6              "intersectionName": "人民路-解放路交叉口",
7              "mode": "FIXED",
8              "currentPhase": "GREEN",
9              "greenDuration": 30,
10             "redDuration": 45,
11             "yellowDuration": 3,
12             "cycleTime": 78
13         }
14     ]
15 }
```

## 9. 调整信号灯配置

**POST** /api/signals/{id}/adjust

请求体:

```
1  {
2      "mode": "FIXED",
3      "greenDuration": 45,
4      "reason": "早高峰车流量增大"
5  }
```

mode可选值: `FIXED` (固定模式) / `INTELLIGENT` (智能模式) / `MANUAL` (手动模式)

## 10. 获取路口所有方向信号灯状态

**GET** /api/multi-direction-traffic/intersections/{intersectionId}/status

响应体:

```
1  {
2      "EAST": {
3          "straightPhase": "RED",
4          "leftTurnPhase": "GREEN",
5          "rightTurnPhase": "GREEN",
6          "straightRemaining": 25,
7          "leftTurnRemaining": 15,
8          "rightTurnRemaining": 15
9      },
10     "SOUTH": {
```

```
11     "straightPhase": "GREEN",
12     "leftTurnPhase": "RED",
13     "rightTurnPhase": "GREEN",
14     "straightRemaining": 30,
15     "leftTurnRemaining": 45,
16     "rightTurnRemaining": 30
17   },
18   "WEST": { ... },
19   "NORTH": { ... }
20 }
```

## 11. 获取交通统计数据

GET /api/dashboard/stats

响应体:

```
1  {
2    "totalVehicles": 12580,
3    "avgSpeed": 35.5,
4    "congestionLevel": "MODERATE",
5    "peakHour": "08:00-09:00"
6  }
```

## 12. 获取热力图数据

GET /api/dashboard/heatmap

响应体:

```
1  [
2    {
3      "intersectionId": "1",
4      "latitude": 31.2304,
5      "longitude": 121.4737,
6      "density": 85
7    }
8  ]
```

## 13. 获取元数据（用于下拉选择）

GET /api/multi-direction-traffic/metadata

响应体:

```
1  {
2    "directions": {
3      "EAST": "东",
4      "SOUTH": "南",
5      "WEST": "西",
6      "NORTH": "北"
7    }
8  }
```

```
7 },
8 "turnTypes": {
9     "STRAIGHT": "直行",
10    "LEFT_TURN": "左转",
11    "RIGHT_TURN": "右转",
12    "U_TURN": "掉头"
13 },
14 "lightPhases": {
15     "RED": "红灯",
16     "YELLOW": "黄灯",
17     "GREEN": "绿灯"
18 },
19 "violationTypes": {
20     "RED_LIGHT": "闯红灯",
21     "ILLEGAL_TURN": "违法转弯",
22     "WRONG WAY": "逆行",
23     "CROSS_SOLID_LINE": "跨实线"
24 }
25 }
```

## 四、给冯的对接文档 (AI算法)

### 对接概述

后端已准备好以下接口供AI大模型算法调用：

1. **上报检测到的违章** - AI检测到违章后调用
2. **违章验证** - 验证检测结果是否符合当前信号灯状态
3. **回调机制** - 异步任务完成后的通知

### 核心接口 1：上报违章检测结果

**POST** /api/violations/report

这是AI检测到违章后调用的主要接口。

请求体：

```
1 {
2     "intersectionId": 1,
3     "direction": "SOUTH",
4     "turnType": "STRAIGHT",
5     "plateNumber": "京A12345",
6     "violationType": "RED_LIGHT",
7     "imageUrl": "https://minio.example.com/violations/frame_001.jpg",
8     "aiConfidence": 0.95,
9     "occurredAt": "2025-12-22T10:30:00"
10 }
```

字段说明：

字段	类型	必填	说明
intersectionId	Long	✓	路口ID (数字或字符串均可, 如 1 或 "intersection-1")
direction	String	✗	方向: EAST / SOUTH / WEST / NORTH
turnType	String	✗	转弯类型: STRAIGHT / LEFT_TURN / RIGHT_TURN / U_TURN
plateNumber	String	✓	车牌号 (如 "京A12345")
violationType	String	✓	违章类型 (见下表)
imageUrl	String	✓	违章截图URL
aiConfidence	Float	✗	AI置信度, 0-1之间, 默认0.95
occurredAt	String	✗	违章发生时间, ISO 8601格式

支持的违章类型 (violationType) :

值	中文	说明
RED_LIGHT	闯红灯	红灯时车辆越过停止线
WRONG WAY	逆行	车辆逆向行驶
CROSS_SOLID_LINE	跨实线/压实线	车辆压实线行驶
ILLEGAL TURN	违法转弯	违规拐弯压实线
SPEEDING	超速	超速行驶

响应体:

```

1  {
2    "id": 123,
3    "message": "Violation reported successfully"
4 }
```

## 核心接口 2: 方向特定的违章检测

POST /api/violation-detection/directions/{direction}/detect-frame

路径参数:

- direction: EAST / SOUTH / WEST / NORTH

请求体:

```
1 {
2     "intersectionId": 1,
3     "plateNumber": "京A12345",
4     "violationType": "RED_LIGHT",
5     "turnType": "LEFT_TURN",
6     "imageUrl": "https://minio.example.com/violations/frame_001.jpg",
7     "aiConfidence": 0.92,
8     "frameTimestamp": "2025-12-22T10:30:00"
9 }
```

响应体:

```
1 {
2     "success": true,
3     "violationId": "550e8400-e29b-41d4-a716-446655440000",
4     "message": "方向特定违章检测处理完成",
5     "direction": "SOUTH",
6     "data": { ... },
7     "timestamp": "2025-12-22T10:30:00"
8 }
```

## 核心接口 3：验证违章是否成立

在上报违章前，可以先调用此接口验证检测结果是否符合当前信号灯状态。

**POST** /api/multi-direction-traffic/intersections/{intersectionId}/validate-violation

请求体:

```
1 {
2     "direction": "SOUTH",
3     "turnType": "STRAIGHT",
4     "violationType": "RED_LIGHT"
5 }
```

响应体:

```
1 {
2     "intersectionId": 1,
3     "direction": "南",
4     "turnType": "直行",
5     "violationType": "RED_LIGHT",
6     "isViolation": true,
7     "message": "构成违章"
8 }
```

验证逻辑说明:

- RED\_LIGHT (闯红灯) : 当前信号灯为红灯时构成违章
- ILLEGAL\_TURN (违法转弯) : 当前信号灯不是绿灯时构成违章
- WRONG WAY (逆行) : 始终构成违章 (与信号灯无关)

- `CROSS_SOLID_LINE` (压实线) : 始终构成违章 (与信号灯无关)

## 核心接口 4：获取当前信号灯状态

在检测违章前，可以获取当前信号灯状态作为参考。

**GET** `/api/multi-direction-traffic/intersections/{intersectionId}/directions/{direction}/turns/{turnType}/status`

路径参数：

- `intersectionId`: 路口ID
- `direction`: EAST / SOUTH / WEST / NORTH
- `turnType`: STRAIGHT / LEFT\_TURN / RIGHT\_TURN

响应体：

```
1 {  
2   "intersectionId": 1,  
3   "direction": "SOUTH",  
4   "directionName": "南",  
5   "turnType": "STRAIGHT",  
6   "turnTypeName": "直行",  
7   "currentPhase": "RED",  
8   "timestamp": "2025-12-22T10:30:00"  
9 }
```

## 核心接口 5：AI回调接口

用于异步任务完成后的通知。

**POST** `/api/violation-detection/ai-callback`

请求体：

```
1 {  
2   "taskId": "550e8400-e29b-41d4-a716-446655440000",  
3   "status": "COMPLETED",  
4   "violations": [  
5     {  
6       "plateNumber": "京A12345",  
7       "violationType": "RED_LIGHT",  
8       "imageUrl": "https://minio.example.com/frame1.jpg",  
9       "confidence": 0.95,  
10      "timestamp": "2025-12-22T10:30:00"  
11    }  
12  ]  
13 }
```

响应体：

```
1 | {
2 |     "success": true,
3 |     "message": "AI回调处理成功",
4 |     "taskId": "550e8400-e29b-41d4-a716-446655440000",
5 |     "timestamp": "2025-12-22T10:30:01"
6 | }
```

## 核心接口 6：AI分析完成通知

**POST** /api/violation-detection/ai-completed

请求体：

```
1 | {
2 |     "taskId": "550e8400-e29b-41d4-a716-446655440000",
3 |     "totalFramesProcessed": 1000,
4 |     "violationsDetected": 5,
5 |     "processingTimeMs": 12500
6 | }
```

## AI对接完整流程示例

```
1 | 1. AI收到视频流/图片帧
2 |     ↓
3 | 2. (可选) 调用 GET /api/multi-direction-traffic/intersections/{id}/status
4 |     获取当前信号灯状态
5 |     ↓
6 | 3. AI模型检测违章（闯红灯、逆行、压实线、违法拐弯）
7 |     ↓
8 | 4. (可选) 调用 POST /api/multi-direction-traffic/intersections/{id}/validate-
9 |     violation
10 |     验证检测结果是否符合当前信号灯状态
11 |     ↓
12 | 5. 调用 POST /api/violations/report 上报违章
13 |     或调用 POST /api/violation-detection/directions/{direction}/detect-frame
14 |     ↓
15 | 6. 后端保存违章记录到MySQL + Redis
16 |     ↓
17 | 7. 后端通过WebSocket推送实时警报给前端
```

## 推荐对接方案

### 方案一：简单模式

直接调用 POST /api/violations/report 上报检测结果，后端会自动处理。

### 方案二：完整模式

1. 先获取信号灯状态
2. 验证违章是否成立
3. 上报违章（包含方向信息）

## 五、功能总结

模块	功能	状态
用户认证	登录/登出/Token验证	✓ 已实现
权限管理	管理员/交警角色区分	✓ 已实现
交警管理	创建/查询/禁用账号	✓ 已实现
违章上报	AI上报违章记录	✓ 已实现
违章查询	分页查询/详情查看	✓ 已实现
违章处理	确认/驳回违章	✓ 已实现
信号灯配置	查询/调整信号灯	✓ 已实现
多方向控制	四方向独立信号控制	✓ 已实现
违章验证	结合信号灯状态验证	✓ 已实现
实时监控	路口状态/热力图/统计	✓ 已实现
WebSocket	实时违章警报推送	✓ 已实现
AI对接	回调/完成通知接口	✓ 已实现