ChatGPT

车辆维修管理系统 RESTful API 文档

本系统采用统一 JSON 返回格式，包括 `code`、`message`、`data` 等字段[1][2]。其中 `code=0` 表示成功，其它非零值表示错误类型；`message` 用于提示信息，`data` 为具体业务数据。鉴权采用 JWT（JSON Web Token）：客户端登录成功后获得 Token，后续调用需在 HTTP 请求头 `Authorization` 中携带该 Token[3]。API 设计遵循 RESTful 原则：所有资源（如用户、车辆、工单等）通过唯一 URI 表示[4]，URI 应使用名词（资源）形式而非动词[5]，并使用 HTTP 标准方法（GET/POST/PUT/DELETE 等）进行增删改查操作。

注: 所有需要登录的接口均需在 `Authorization: Bearer <JWT>` 请求头中传入合法 JWT 令牌[3]；管理员接口则在此基础上要求用户具有管理员权限。

用户与认证

- **POST /users/register**
- **说明**: 用户注册，创建新账号。
- **权限**: 无需登录。
- **请求参数** (JSON Body):

```
{
  "username": "string",  // 用户名，必填
  "password": "string",  // 密码，必填
  "phone": "string"      // 手机号，可选
}
```

- **响应示例** (成功):

```
{
  "code": 0,
  "message": "注册成功",
  "data": {
    "userId": 123,
    "username": "alice",
    "phone": "1234567890"
  }
}
```

- **POST /users/login**

- **说明**: 用户登录，验证身份并返回 JWT 令牌。
- **权限**: 无需登录。
- **请求参数** (JSON Body):

```
{
  "username": "string",  // 用户名或邮箱
```

```
  "password": "string"   // 密码
}
```

- **响应示例** (成功):

```
{
  "code": 0,
  "message": "登录成功",
  "data": {
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6..."  // JWT 令牌
  }
}
```

- **GET /users/me**

- **说明**: 获取当前登录用户的信息。
- **权限**: 需登录。
- **请求参数**: 无（使用 Header 中的 JWT）。

- **响应示例**:

```
{
  "code": 0,
  "message": "",
  "data": {
    "userId": 123,
    "username": "alice",
    "phone": "1234567890",
    "role": "user"  // 用户角色，如 user/technician/admin
  }
}
```

- **PUT /users/me**

- **说明**: 更新当前用户信息（如联系方式等）。
- **权限**: 需登录。
- **请求参数** (JSON Body):

```
{
  "phone": "string",   // 新手机号，可选
  "password": "string"  // 修改密码，可选
}
```

- **响应结构**: `{ code, message, data: null }`，`code=0` 表示更新成功。

- **GET /users/{id}**

- **说明**: 按用户 ID 查询用户信息。普通用户只能查询自身信息（即 `{id}` 等于自己 ID），管理员可查询任意用户。
- **权限**: 需登录；管理员可用。
- **请求参数**: 路径参数 `id` （int）。
- **响应结构**: 同 **GET /users/me**。

车辆管理

- **POST /vehicles**
- **说明**: 登记新车辆。
- **权限**: 需登录。
- **请求参数** (JSON Body):

```
{
 "plateNumber": "string",  // 车牌号，必填
 "model": "string",      // 车型，必填
 "year": 2020,          // 出厂年份，可选
 "ownerId": 123         // 车主用户ID（可从 JWT 中获取）
}
```

- **响应示例**:

```
{
 "code": 0,
 "message": "登记成功",
 "data": {
  "vehicleId": 456,
  "plateNumber": "ABC-123",
  "model": "Sedan",
  "year": 2020,
  "ownerId": 123
 }
}
```

- **GET /vehicles**

- **说明**: 查询车辆列表。普通用户返回其拥有的车辆，管理员可查询所有车辆。支持分页和过滤（如按车牌号、用户ID）。
- **权限**: 需登录。
- **请求参数** (Query):
  - `ownerId` (int，非必填)：指定用户ID过滤车辆；
  - `page`、`size` 等分页参数。
- **响应示例**:

```
{
 "code": 0,
 "message": "",
```

```
  "data": [
   {
    "vehicleId": 456,
    "plateNumber": "ABC-123",
    "model": "Sedan",
    "year": 2020,
    "ownerId": 123
   },
   ...
  ]
}
```

- **GET /vehicles/{id}**

- **说明**: 按车辆ID查询车辆详情。普通用户只能查询自己名下的车辆，管理员可查询任意车辆。
- **权限**: 需登录。
- **请求参数**: 路径参数 `id` （int）。

- **响应结构**: 详见 **GET /vehicles** 中单个车辆格式。

- **PUT /vehicles/{id}**

- **说明**: 修改车辆信息（如更新车型、年份）。
- **权限**: 需登录；普通用户仅能修改自己名下车辆，管理员可修改任意。
- **请求参数** (JSON Body): 修改字段，如

```
{
  "model": "string",
  "year": 2021
}
```

- **响应结构**: `{ code, message, data: null }`。

- **DELETE /vehicles/{id}**

- **说明**: 删除车辆。
- **权限**: 需登录；普通用户仅能删除自己名下车辆，管理员可删除任意。
- **请求参数**: 路径参数 `id`。
- **响应结构**: `{ code, message, data: null }`。

维修工单管理

- **POST /orders**
- **说明**: 提交维修工单。用户提交维修请求，系统生成新的工单（状态为“待分配”）。
- **权限**: 需登录。
- **请求参数** (JSON Body):

```
{
  "vehicleId": 456,      // 车辆ID
```

```
  "description": "string", // 维修问题描述，必填
  "appointmentTime": "2025-06-01T10:00:00", // 预约上门时间，可选
  "estimatedCost": 500    // 预估费用，可选
}
```

- **响应示例**:

```
{
  "code": 0,
  "message": "提交成功",
  "data": {
    "orderId": 789,
    "status": "PENDING",
    "vehicleId": 456,
    "description": "刹车异响",
    "appointmentTime": "2025-06-01T10:00:00",
    "estimatedCost": 500
  }
}
```

- **GET /orders**

- **说明**: 查询工单列表。普通用户返回自己提交的工单，维修员返回自己接受的工单，管理员返回所有工单。支持分页及按状态、用户等过滤。
- **权限**: 需登录。
- **请求参数** (Query):
  - `status` (string)：按工单状态过滤（如 PENDING/ASSIGNED/IN_PROGRESS/COMPLETED/CANCELLED）；
  - `ownerId` (int)：按用户ID过滤；
  - `page`、`size` 等分页参数。
- **响应示例**:

```
{
  "code": 0,
  "message": "",
  "data": [
    {
      "orderId": 789,
      "status": "PENDING",
      "vehicleId": 456,
      "description": "刹车异响",
      "appointmentTime": "2025-06-01T10:00:00",
      "estimatedCost": 500
    },
    ...
  ]
}
```

- **GET /orders/{id}**

- **说明**: 查询指定工单详情，包括当前状态、分配的维修员、维修记录等。普通用户仅能查看自己提交的工单，维修员仅能查看自己接收的工单，管理员可查看任意。
- **权限**: 需登录。
- **请求参数**: 路径参数 `id`（工单ID）。

- **响应结构**:

```
{
 "code": 0,
 "message": "",
 "data": {
  "orderId": 789,
  "status": "ASSIGNED",
  "vehicleId": 456,
  "description": "刹车异响",
  "appointmentTime": "2025-06-01T10:00:00",
  "estimatedCost": 500,
  "assignedTo": 234,        // 维修员用户ID
  "serviceFee": 0,
  "materials": [],
  "history": [...]        // 维修记录日志
 }
}
```

- **PUT /orders/{id}**

- **说明**: 更新工单信息。主要用于订单取消或补充信息等操作（状态字段可设为 “CANCELLED” 取消工单）。
- **权限**: 需登录；普通用户仅能取消自己的工单（状态改为 CANCELLED），管理员可修改任意工单信息。
- **请求参数** (JSON Body): 如

```
{
 "status": "CANCELLED"
}
```

- **响应结构**: `{ code, message, data: null }`。

- **POST /orders/{id}/assign**

- **说明**: 指定维修员接单（手工分配）。管理员可调用此接口将工单分配给某个维修员。系统也可以后台自动分配满足条件的维修员。
- **权限**: 管理员。
- **请求参数** (JSON Body):

```
{
 "technicianId": 234  // 指定的维修员用户ID
}
```

- **响应结构**: `{ code, message, data: null }`。

- **POST /orders/{id}/accept**

- **说明**: 维修员接单。维修员调用此接口接受系统分配或自选的工单，工单状态变为 `ASSIGNED`。
- **权限**: 需登录（维修员）。
- **请求参数**: 无（仅路径参数 `id`）。
- **响应结构**: `{ code, message, data: null }`。

- **GET /orders/assigned**

- **说明**: 维修员查看自己已接收的工单列表（状态为 ASSIGNED 或 IN_PROGRESS）。
- **权限**: 需登录（维修员）。
- **请求参数**: 可选分页。
- **响应结构**: 类似 **GET /orders** 的列表格式。

维修记录与收入查询

- **POST /orders/{id}/records**
- **说明**: 维修员提交维修记录，包括所用材料和工时费用。调用后工单状态自动更新为 `COMPLETED`。
- **权限**: 需登录（维修员）。
- **请求参数** (JSON Body):

```
{
 "description": "更换机油，清洗滤芯",
 "materials": [
  {"name":"机油","quantity":1,"unitCost":100},
  {"name":"空气滤芯","quantity":1,"unitCost":50}
 ],
 "laborHours": 2,
 "laborCost": 200
}
```

- **响应结构**: `{ code, message, data: null }`（工单状态更新为 `COMPLETED`，并记录下维修日志）。

- **GET /orders/{id}/records**

- **说明**: 查询某工单的所有维修记录日志。包括每次维修的描述、材料和费用。
- **权限**: 需登录；普通用户只能查自己工单的记录，维修员只能查自己工单的记录，管理员可查任何。
- **请求参数**: 路径参数 `id`。

- **响应示例**:

```
{
 "code": 0,
 "message": "",
 "data": [
  {
   "description": "更换机油，清洗滤芯",
   "materials": [ ... ],
   "laborHours": 2,
   "laborCost": 200,
   "timestamp": "2025-06-01T15:30:00"
  },
  ...
 ]
}
```

- **GET /mechanics/{id}/income**

- **说明**: 查询指定维修员的工时收入统计。返回该维修员所有完成工单的总工时和总收入。
- **权限**: 需登录；维修员本人或管理员可查询。
- **请求参数**: 路径参数 `id`（维修员用户ID）。
- **响应示例**:

```
{
 "code": 0,
 "message": "",
 "data": {
  "totalHours": 120,
  "totalIncome": 6000
 }
}
```

预约管理

- **POST /appointments**
- **说明**: 创建维修预约。用户可预约上门服务时间。
- **权限**: 需登录。
- **请求参数** (JSON Body):

```
{
 "vehicleId": 456,
 "appointmentTime": "2025-06-01T10:00:00",
 "serviceType": "洗车"        // 服务类型或备注
}
```

- **响应示例**:

```
{
 "code": 0,
 "message": "预约成功",
 "data": {
  "appointmentId": 321,
  "vehicleId": 456,
  "appointmentTime": "2025-06-01T10:00:00",
  "serviceType": "洗车"
 }
}
```


- **GET /appointments**

- **说明**: 查询预约列表。普通用户返回自己所有预约，管理员可查询所有预约。支持分页及按日期、用户过滤。
- **权限**: 需登录。
- **请求参数** (Query): `userId`（可选）、`date`（预约日期）等。

- **响应示例**:


```
{
 "code": 0,
 "message": "",
 "data": [
  {
   "appointmentId": 321,
   "vehicleId": 456,
   "appointmentTime": "2025-06-01T10:00:00",
   "serviceType": "洗车"
  },
  ...
 ]
}
```


- **GET /appointments/{id}**

- **说明**: 查询指定预约详情。普通用户只能查看自己的预约。
- **权限**: 需登录。
- **请求参数**: 路径参数 `id`。

- **响应结构**: 同 **GET /appointments** 中单个预约格式。

- **PUT /appointments/{id}**

- **说明**: 修改预约信息（如更改时间、服务类型）。
- **权限**: 需登录；普通用户仅能修改自己的预约。
- **请求参数** (JSON Body):

```
{
 "appointmentTime": "2025-06-02T14:00:00",
 "serviceType": "保养"
}
```

- **响应结构**: `{ code, message, data: null }`。

- **DELETE /appointments/{id}**

- **说明**: 取消预约。
- **权限**: 需登录；普通用户仅能取消自己的预约。
- **请求参数**: 路径参数 `id`。
- **响应结构**: `{ code, message, data: null }`。

评价管理

- **POST /orders/{id}/feedback**
- **说明**: 提交服务反馈（评价）。用户在工单完成后对维修服务进行评价。
- **权限**: 需登录；只能对自己提交的且已完成的工单评价。
- **请求参数** (JSON Body):

```
{
 "rating": 5,        // 评分，1-5
 "comment": "string"    // 文字评价
}
```

- **响应结构**: `{ code, message, data: null }`。

- **GET /orders/{id}/feedback**

- **说明**: 查询指定工单的反馈信息（包括用户评价）。
- **权限**: 需登录；普通用户仅能查看自己工单的评价，维修员或管理员可查看任意。
- **请求参数**: 路径参数 `id`（工单ID）。

- **响应示例**:

```
{
 "code": 0,
 "message": "",
 "data": {
  "orderId": 789,
  "rating": 5,
  "comment": "服务很满意",
  "submittedAt": "2025-06-02T12:00:00"
 }
}
```

- **GET /mechanics/{id}/feedback**

- **说明**: 查询某维修员的所有收到的评价。
- **权限**: 需登录；维修员本人或管理员可查看。
- **请求参数**: 路径参数 `id`（维修员ID）。
- **响应示例**:


```
{
 "code": 0,
 "message": "",
 "data": [
  {
   "orderId": 789,
   "rating": 5,
   "comment": "服务很满意"
  },
  ...
 ]
}
```


管理员接口

- **GET /admin/statistics**
- **说明**: 系统监控与数据统计，返回用户数、车辆数、订单总数、总收入等汇总信息。
- **权限**: 管理员。
- **请求参数**: 无。

- **响应示例**:


```
{
 "code": 0,
 "message": "",
 "data": {
  "totalUsers": 200,
  "totalVehicles": 150,
  "totalOrders": 120,
  "totalRevenue": 50000
 }
}
```


- **GET /admin/users**

- **说明**: 获取所有用户列表，用于后台管理。
- **权限**: 管理员。
- **请求参数** (可选分页)。

- **响应示例**: 列表形式，结构同 **GET /users**。

- **GET /admin/vehicles**

- **说明**: 获取所有车辆列表，后台查看。
- **权限**: 管理员。
- **响应示例**: 列表形式，结构同 **GET /vehicles**。

- **GET /admin/orders**

- **说明**: 获取所有工单列表，后台管理。
- **权限**: 管理员。
- **请求参数** (可选状态过滤)。
- **响应示例**: 列表形式，结构同 **GET /orders**。

- **GET /admin/appointments**

- **说明**: 获取所有预约列表，后台管理。
- **权限**: 管理员。
- **响应示例**: 列表形式，结构同 **GET /appointments**。

- **GET /admin/feedbacks**

- **说明**: 获取所有服务反馈列表，后台管理。
- **权限**: 管理员。
- **请求参数** (可按评分等过滤)。
- **响应示例**: 列表形式，包括 `orderId`、`rating`、`comment` 等字段。

以上接口均采用统一的 JSON 响应格式 `{code, message, data}`，方便前端统一解析；其中 **code=0** 表示成功，其它值为不同错误类型 [1] [2] 。有关 RESTful 设计原则，请参考相关资料 [4] [5] 。

---

[1] Spring Boot returns the results, global abnormalities, verification - Programmer Sought
https://www.programmersought.com/article/474310851668/

[2] 01----------统一封装返回信息{code,msg,data}_result封装code,msg,data-CSDN博客
https://blog.csdn.net/qq_33296156/article/details/82078654

[3] API Endpoints Protection Using JWT
https://www.netguru.com/blog/api-endpoints-protection-jwt

[4] [5] Web API Design Best Practices - Azure Architecture Center | Microsoft Learn
https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design