汽车维修系统

1. 背景和意义
2. 开发技术

前端：

Vue.js

后端：

C#：主体

Signal：实时实现消息推送

Dapper：数据库访问

1. 需求分析

## 项目介绍

该系统设计的目的是对汽车报修保养和配件仓库库存进行电子化管理，主要实现汽车预约，维修登记，库存出入库，盘点记录与分和库存自动报警等功能。本系统可提供有关汽车维修，人员分配和库存流动的准确信息，增强作业的合理性，准确性和快捷性，便利维修厂统计和分析员工分配合理性，并可以准确对配件进货，减少库存积压，也让管理员能及时发现库存丢失、数据错误等情况，最大程度地节约成本。

项目总体功能结构图，如图3-1所示。



图3-1 功能结构图

## 系统功能描述

从项目介绍中得到该系统划分为五大模块。分别为一用户模块，有用户登录，注册，注销，个人中心，权限管理和角色管理功能；二维修模块，主要用来车主预约维修和保养；三客户模块，该模块要来记录维修厂合作汽车公司，方便维修模块的选择操作；四库存模块，有库存单元的管理，入库管理，出库管理和盘点管理；五定时任务模块，有入库单删除定时任务，出库单删除的定时任务，盘点单删除的定时任务，消息通知删除的定时任务和库存记录删除的定时任务。

### 用户模块

用户模块需要用户先登录才能操作本系统，登录之后页面跳转到主页面，可以手动选择注销退出。

表3-1 用户模块功能表

|  |  |  |
| --- | --- | --- |
| **模块** | **功能** | **功能描述** |
| 用户模块 | 登录 | 系统用户登录 |
| 注销 | 用户退出系统 |
| 注册 | 注册本系统用户 |
| 个人中心 | 可以在这里进行用户个人信息修改，包括基础信息修改，密码修改和头像修改（待定） |
| 权限管理 | 权限添加和维护 |
| 角色管理 | 角色添加维护和角色赋予 |

### 维修模块

该模块用于司机预约，维修记录和打印签字。

表3-2 基本信息功能表

|  |  |  |
| --- | --- | --- |
| **模块** | **功能** | **功能描述** |
| 维修模块 | 保修预约 | 司机预约 |
| 维修管理 | 维修单的增删改查 |

### 客户模块

该模块要来记录维修厂的合作公司，以方便后续操作。

表3-3 客户模块功能表

|  |  |  |
| --- | --- | --- |
| **模块** | **功能** | **功能描述** |
| 客户模块 | 客户管理 | 对客户进行添加和维护 |

### 库存模块

库存模块可以添加和查询各仓库库元信息，也可以进行库存数量，警报值等信息更新，每次库元信息操作会生成操作记录，方便对库元信息追查；该模块还有入库管理和出库管理功能。

表3-4 库存模块功能表

|  |  |  |
| --- | --- | --- |
| **模块** | **功能** | **功能描述** |
| 库存模块 | 库存查看 | 查看仓库库元信息 |
| 库元修改 | 库元信息修改 |
| 库元添加 | 仓库库元添加 |
| 库元编辑记录查看 | 查看库元修改记录 |
| 入库管理 | 入库单的添加和维护 |
| 出库管理 | 出库单的添加和维护 |
|  | 盘点管理 | 盘点单的添加和维护 |

### 定时任务模块

定时任务模块有入库单删除定时任务，出库单删除的定时任务，盘点单删除的定时任务，消息通知删除的定时任务，库存记录删除和维修单删除的定时任务。

表3-7 定时任务模块功能表

|  |  |  |
| --- | --- | --- |
| **模块** | **功能** | **功能描述** |
| 定时任务模块 | 入库单定时删除 | - |
| 出库单定时删除 | - |
| 盘点单定时删除 | - |
| 通知记录定时删除 | - |
| 维修单定时删除 | - |

## 系统用例模型

在本系统设计中，该系统一开始拥有多种角色，即超级管理员，用户管理员，定时任务管理员，库存管理员，库存入库管理员，库存出库管理员，库存盘点管理员，预约管理员，维修管理员和普通用户。超级管理员拥有系统全部权限，可以操作每个模块功能。也可以在角色管理模块自定义角色，或者在角色管理模块修改上述角色权限；

超级管理员用例图如图3-2所示。



用户管理员用例图

定时任务管理员



库存管理员



库存入库管理员



库存出库管理员（类似入库管理员）

库存盘点管理员（类似入库管理员）

预约管理员（类似入库管理员）

维修管理员（类似入库管理员）

客户管理员



普通用户角色为用户刚注册后的角色，拥有功能较少。普通用户角色用例图如图3-3 所示。



## 系统业务流程

利用活动图使系统各模块业务更加清晰。

### 用户模块

用户填写登录信息，点击登录，系统自动检查数据格式是否符合，不符合提示用户修改，符合再获取数据库数据并比较数据是否一致，若账号密码一致，则页面添转到首页，若不一致，则提示用户重新填写。



图3-12 用户模块活动图

### 客户模块

1. 客户添加活动，管理员填写客户信息，点击添加按钮，系统将客户信息传到后台服务器，查询数据库客户表，并根据查询数据判断提交信息是否有效。客户添加活动如图3-25所示。



图3-25 客户添加活动图

1. 客户更新活动，管理员填写客户更新信息，点击更新按钮，系统将客户信息提交给后台，后台系统将数据添加进数据库，成功保存后返回成功提示信息。客户更新活动图如图3-26所示。



图3-26客户更新活动图

1. 客户删除活动，管理员点击删除按钮，系统提示是否确认删除信息，管理员再次确认，系统根据用户选择的数据，将对应数据库中的数据删除，然后返回删除成功信息。客户删除活动图如图3-27所示。



图3-27 客户删除活动图

### 库存模块

1. 库存添加活动，管理员填写库存信息，点击添加按钮，系统将库存传到后台服务器，查询数据库库存表，并根据查询数据判断提交信息是否有效。库存添加活动如图3-28所示。



图3-28 库存添加活动图

1. 库存更新活动，管理员填写库存更新信息，点击更新按钮，系统将库存信息提交给后台，并查询数据库库存表，并根据查询数据判断提交信息是否有效。库存更新活动图如图3-29所示。



图3-29库存更新活动图

1. 库存删除活动，管理员点击删除按钮，系统提示是否确认删除信息，管理员再次确认，系统根据用户选择的数据，将对应数据库中的数据删除，然后返回删除成功信息。库存删除活动图如图3-30所示。



图3-30 库存删除活动图

### 盘点模块

1. 盘点单添加活动，管理员填写盘点单信息，点击添加按钮，系统将盘点单传到后台服务器，并查询数据库盘点表，并根据查询数据判断提交信息是否有效。盘点单添加活动如图3-33所示。



图3-33 盘点单添加活动图

1. 盘点更新活动，管理员填写盘点更新信息，点击更新按钮，系统将盘点信息提交给后台，查询数据库盘点表，并根据查询数据判断提交信息是否有效。盘点单更新活动图如图3-34所示。



图3-34盘点单更新活动图

### 定时任务模块

定时任务设置活动，用户填写定时信息，确认设置，系统审核用户填写的定时信息，合理更新数据库，并返回成功信息；不合理，返回提示信息。定时任务设置活动如图3-35所示。



图3-35 定时任务设置活动图

定时任务关闭活动，管理员点击关闭按钮，系统提示是否关闭，管理员确认关闭，系统更新数据库定时任务数据信息。定时任务关闭活动图如图3-36所示。



图3-36盘点单更新活动图

维修模块

1. 添加预约
2. 维修单管理

消息推送

1. 库存报警
2. 工具未归还提醒

# 概要设计

## 4.1 逻辑分层结构

本平台采用.NetCode整合Dapper完成系统后端服务器架构，前端使用Vue.js结合Element UI完成页面快速搭建，数据库使用SQL Server。本系统的技术架构图如图4-1所示。



图4-1 技术架构图

下面对每层及技术进行简要说明。

（1）视图层，本平台采用B/S结构(Browser/Server,浏览器/服务器结构)，需要通过浏览器来对本平台实现访问操作。于本层采用了Vue.js结合Element UI 实现页面整体效果，并通过使用request请求将视图层请求数据传递到控制层。

（2）控制层，这里由.Net Code Web Mvc 实现对访问路径的权限控制和路径跳转，可以通过Service接口调用业务逻辑层方法。

（3）业务逻辑层，本层通过调用数据持久层的接口完成对数据库SQL Server的操作。

（4）数据持久层，本层使用Dapper作为数据库的连接框架，并由Dao接口和Dao实现类实现对数据库的操作。

（5）基础设施层，使用SQL Server作为数据库，存储系统数据。

系统的配置信息原理图如图 4-2 所示。



图4-2 系统的配置信息原理图

## 4.2 包图

### 4.2.1 包结构设计

本系统是横向规划，每层建一些包：

（1）视图层包：存放模块的页面；可配合css包、js包等，分别存放页面样式文件和JavaScript代码文件。

（2）控制层包：存放模块的所有Controller类。

（3）业务逻辑包：存放模块的业务接口和业务实现类。

（4）数据访问包：存放Mapper接口和Mapper.xml文件。

（5）公共组件包：存放一些公共组件。

（6）bean包：存放系统实体类。

### 4.2.2 包设计

# 数据库设计

## 概念模型设计

本系统对表的划分可以分为两大类，一类是用户模块涉及的用户关系表，主要表为用户表，角色表和权限表；另一类是库存管理涉及的关系表，主要包括基础商品表，库存表以及对库存操作的各种记录表。具体实体-关系（E-R）图如图5-1所示。

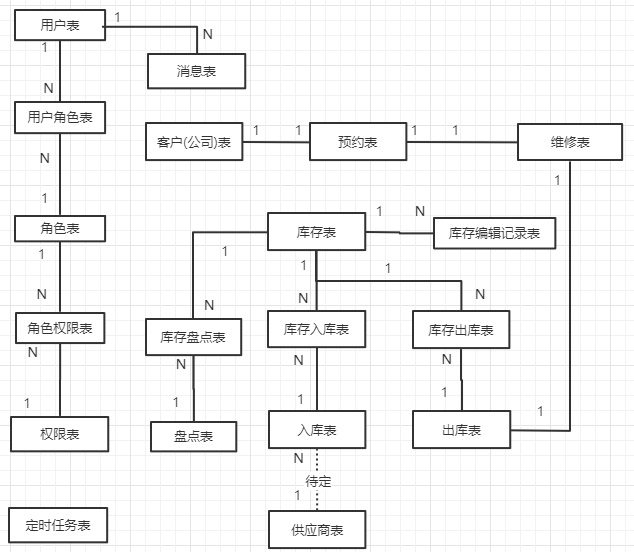


图5-1 实体-关系（E-R）图

## 数据库表设计

（1）用户表，该表用来存储本系统用户的基本信息，包括账号，密码，手机和邮箱等。

表5-1 用户表（ums\_user）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| id | int |  | NOT NULL | 是 | 用户账号唯一标识 | 是 |
| UserName | varchar | 255 | NOT NULL |  | 用户名 | 是 |
| NickName | varchar | 255 | NULL |  | 昵称 | 否 |
| Phone | varchar | 255 | NULL |  | 用户手机号 | 否 |
| Email | varhcar | 255 | NULL |  | 用户邮箱 | 否 |

续表5-1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| Photo | varchar | 255 | NULL |  | 用户头像链接 | 否 |
| Sex | varchar | 4 | NULL |  | 用户性别，’0’为女，’1’为男 | 否 |
| Pwd | varchar | 255 | NOT NULL |  | 账号密码 | 否 |

（2）用户角色表，该表用来记录用户和角色的对应关系。

表5-2 用户角色表（ums\_user\_role）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| id | int |  | NOT NULL | 是 | 唯一标识 | 是 |
| UserId | int |  | NOT NULL |  | 用户ID | 否 |
| RoleId | int |  | NOT NULL |  | 角色ID | 否 |

（3）角色表，该表用来存储管理员创建的角色信息，字段为角色ID，角色名和角色描述。

表5-3 角色表（ums\_role）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| id | int | 11 | NOT NULL | 是 | 唯一标识 | 是 |
| RoleName | varchar | 255 | NOT NULL |  | 角色名 | 否 |
| Description | varchar | 255 | NULL |  | 描述 | 否 |

（4）角色权限表，该表用来记录角色和权限的对应关系。

表5-4 角色权限表（ums\_role\_authority）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| id | int | 11 | NOT NULL | 是 | 唯一标识 | 是 |
| AuthorityId | int | 11 | NOT NULL |  | 权限ID | 否 |
| RoleId | int | 11 | NOT NULL |  | 角色ID | 否 |

（5）权限表，该表用来存储超级管理员创建的权限信息。

表5-5 权限表（ums\_authority）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| id | int | 11 | NOT NULL | 是 | 唯一标识 | 是 |
| AuthorityName | varchar | 255 | NOT NULL |  | 权限名 | 否 |
| Description | varchar | 255 | NULL |  | 描述 | 否 |
| Icon | varchar | 255 | NULL |  | 菜单图标 | 否 |
| Resource | varchar | 255 | NULL |  | 跳转路径 | 否 |
| Level | varchar | 255 | NOT NULL |  | 菜单级别 | 否 |

（6）消息表，该表用来记录库存报警信息和各种通知消息，主要字段为标题和内容。

表5-6 消息表（nms\_notification）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| id | int | 11 | NOT NULL | 是 | 唯一标识 | 是 |
| Title | varchar | 255 | NOT NULL |  | 消息标题 | 否 |
| Content | varchar | 255 | NOT NULL |  | 消息内容 | 否 |
| CreateTime | varchar | 255 | NOT NULL |  | 创建时间 | 否 |

（7）用户消息表，该表用来记录用户和消息的关联关系。

表5-7 用户消息表（nms\_user\_notification）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| NotificationId | int | 11 | NOT NULL |  | 消息ID | 否 |
| UserId | int | 11 | NOT NULL |  | 用户ID | 否 |
| Status | varchar | 255 | NOT NULL |  | ‘1’为已查看，‘0’没有查看 | 否 |

（8）客户表，该表用来储存维修厂客户公司信息，主要字段有客户名，联系人和手机号等。

表5-13 客户表（cms\_client）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| id | int | 11 | NOT NULL | 是 | 客户ID | 是 |
| Company | varchar | 255 | NOT NULL |  | 客户名 | 是 |
| Address | varchar | 255 | NOT NULL |  | 客户地址 | 否 |
| Phone | varchar | 255 | NOT NULL |  | 手机号 | 否 |
| Contact | varchar | 255 | NOT NULL |  | 联系人 | 否 |
| email | varchar | 255 | NOT NULL |  | 邮箱 | 否 |
| TYPE | Varchar |  |  |  | 类型 |  |
| DECRIPTION | Varchar |  |  |  | 描述 |  |

（18）盘点表，该表用来记录盘点单信息。

表5-18 盘点表（sms\_check）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| id | int | 11 | NOT NULL | 是 | ID | 是 |
| CheckNo | varchar | 255 | NOT NULL |  | 盘点单号 | 是 |
| Room | int | 255 | NOT NULL |  | 房间号 | 否 |
| Operator | varchar | 255 | NOT NULL |  | 操作员 | 否 |
| CheckDate | varchar | 255 | NOT NULL |  | 盘点日期 | 否 |
| TotalPrice | decimal | 10 | NULL |  | 总金额 | 否 |
| Status | varchar | 30 | NOT NULL |  | 状态 | 否 |

（20）入库表，该表用来记录入库单信息。

表5-20 入库表（sms\_entry）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| id | int | 11 | NOT NULL | 是 | ID | 是 |
| EntryNo | varchar | 255 | NOT NULL |  | 入库单号 | 是 |
| Room | Varchar | 255 | NOT NULL |  | 房间 | 否 |
| Operator | varchar | 255 | NOT NULL |  | 操作员 | 否 |
| EntryDate | varchar | 255 | NOT NULL |  | 入库日期 | 否 |
| TotalPrice | decimal | 10 | NULL |  | 总金额 | 否 |
| Batch | int | 4 | NOT NULL |  | 批次 | 否 |
| SupplierId | int | 11 | NOT NULL |  | 供应商ID | 否 |
| Desciprion | varchar | 255 | NULL |  | 备注 | 否 |

（21）出库表，该表用来记录出库单信息。

表5-21 出库表（sms\_out）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| id | int | 11 | NOT NULL | 是 | ID | 是 |
| OutNo | varchar | 255 | NOT NULL |  | 出库单号 | 是 |
| Room | Varchar | 255 | NOT NULL |  | 房间 | 否 |
| Operator | varchar | 255 | NOT NULL |  | 操作员 | 否 |
| OutDate | varchar | 255 | NOT NULL |  | 出库日期 | 否 |
| TotalPrice | decimal | 10 | NULL |  | 总金额 | 否 |
| Batch | int | 4 | NOT NULL |  | 批次 | 否 |
| Description | varchar | 255 | NULL |  | 备注 | 否 |

（22）定时任务表，该表用来记录定时任务信息。

表5-22 定时任务表（jms\_scheduler）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| id | int |  | NOT NULL | 是 | 主键 | 是 |
| JobName | varchar | 255 | NOT NULL |  | 任务名 | 是 |
| CronStr | varchar | 255 | NOT NULL |  | 执行时间字符串 | 否 |
| Limit | varchar | 255 | NOT NULL |  | 删除范围 | 否 |
| Status | varchar | 255 | NOT NULL |  | 状态，‘0’为关闭中，‘1’已经开启 | 否 |

（23）库存表，该表用来储存库存信息。

表5-23 库存表（sms\_sku）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| id | int | 11 | NOT NULL | 是 | ID | 是 |
| Name | varchar | 255 | NOT NULL |  | 库存单元名 | 否 |
| Room | varchar | 30 | NOT NULL |  | 房间 | 否 |
| Self | Varchar | 30 | NOT NULL |  | 货架 | 否 |
| Brand | varchar | 255 | NOT NULL |  | 品牌 | 否 |
| Price | Decimal | 11 | NOT NULL |  | 单价 | 否 |
| Unit | int | 11 | NOT NULL |  | 单位 | 否 |
| Quantity | int | 11 | NOT NULL |  | 数量 | 否 |
| Alarm | int | 11 | NOT NULL |  | 警报值 | 否 |
| Description | varchar | 255 | NULL |  | 描述 | 否 |
| Type | varchar | 255 | NULL |  | 规格 | 否 |
| Status | int |  | NOT NULL |  | 记录新旧，1为旧 | 否 |
| OldPartId | int |  | NULL |  | 旧配件来源 | 否 |
| SkuNo | Varchar |  | NOTNULL |  | 配件编号 | 否 |

（24）库存盘点表，该表用来关联库存与盘点单关系。主要字段有盘点数量，盘点金额，相差数量，相差金额等。

表5-24 库存盘点表（sms\_sku\_check）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| CheckId | int | 11 | NOT NULL |  | 盘点单ID | 否 |
| SkuId | varchar | 255 | NOT NULL |  | 库存ID | 否 |
| CheckQuantity | varchar | 255 | NOT NULL |  | 盘点数量 | 否 |
| CheckTotalPrice | decimal | 255 | NOT NULL |  | 盘点金额 | 否 |
| DifferenceNum | varchar | 255 | NOT NULL |  | 相差数量 | 否 |
| DifferencePrice | decimal | 10 | NOT NULL |  | 相差金额 | 否 |
| Remark | varchar | 255 | NULL |  | 备注 | 否 |
| Price | decimal | 10 | NOT NULL |  | 单价 | 否 |
| Quantity | int | 11 | NOT NULL |  | 账面数量 | 否 |
| TotalPrice | decimal | 10 | NOT NULL |  | 账面金额 | 否 |

（25）库存修改表，该表用来记录库存修改记录信息。

表5-25 库存修改表（sms\_sku\_edit）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| id | int | 11 | NOT NULL | 是 | ID | 是 |
| Room | varchar | 255 | NOT NULL |  | 房间 | 是 |
| SkuId | varchar | 255 | NOT NULL |  | 库存ID | 否 |
| OldQuantity | varchar | 255 | NOT NULL |  | 原来数量 | 否 |
| OldAlarm | int | 11 | NOT NULL |  | 原来警报值 | 否 |
| OldPrice | decimal | 10 | NULL |  | 原来单价 | 否 |
| NewQuantity | int | 11 | NOT NULL |  | 新的数量 | 否 |
| NewPrice | decimal | 11 | NULL |  | 新的单价 | 否 |
| NewAlarm | int | 11 | NOT NULL |  | 新的警报值 | 否 |
| Remark | varchar | 255 | NOT NULL |  | 备注 | 否 |
| EditDate | varchar | 255 | NOT NULL |  | 修改日期 | 否 |
| Type | varchar | 255 | NOT NULL |  | ‘0’为删除，‘1’为修改 | 否 |
| Operator | varchar | 255 | NOT NULL |  | 操作账号 | 否 |

（26）库存入库表，该表用来记录库存与入库表的关联关系，及储存入库库存的数量和金额等信息。

表5-26 库存入库表（sms\_sku\_entry）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| EntryId | int | 11 | NOT NULL |  | 入库单ID | 否 |
| SkuId | int | 11 | NOT NULL |  | 库存ID | 否 |
| Quantity | int | 11 | NOT NULL |  | 入库数量 | 否 |
| Price | decimal | 10 | NOT NULL |  | 入库单价 | 否 |
| TotalPrice | decimal | 10 | NOT NULL |  | 总金额 | 否 |
| Status | int |  | NULL |  | 1为旧 | 否 |
| OldPartId | int |  | NULL |  | 旧配件来源 | 否 |

（27）库存出库表，该表用来记录库存与出库表的关联关系，及储存出库库存的数量和金额等信息。

表5-27 库存出库表（sms\_sku\_out）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| OutId | int | 11 | NOT NULL |  | 出库单ID | 否 |
| SkuId | int | 11 | NOT NULL |  | 库存ID | 否 |
| Quantity | int | 11 | NOT NULL |  | 出库数量 | 否 |
| Price | decimal | 10 | NOT NULL |  | 出库单价 | 否 |
| TotalPrice | decimal | 10 | NOT NULL |  | 总金额 | 否 |
| Type | int |  | NOT NULL |  | 是否是工具，1是配件，0是工具 | 否 |

（28）预约表，该表用来记录保修信息。

表5-27 预约表（mms\_appointment）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| Id | int | 11 | NOT NULL | 是 | 主键 | 是 |
| AppointmentNo | varchar | 30 | NOT NULL |  | 预约编号 | 否 |
| CompanyId | int | 11 | NOT NULL |  | 客户公司 | 否 |
| CarLicense | varchar | 20 | NOT NULL |  | 车牌号 | 否 |
| Description | varchar | 1024 | NOT NULL |  | 问题描述 | 否 |
| AppointedDate | Date |  | NOT NULL |  | 预约日期 | 否 |
| Type | varchar | 30 | NOT NULL |  | 车型 | 否 |
| OCU | Varchar | 30 | NULL |  | 创建账号 | 否 |
| OCD | Date |  | NULL |  | 创建时间 | 否 |
| LUC | Varchar | 30 | NULL |  | 最后更新账号 | 否 |
| LUD | Date |  | NULL |  | 最后更新时间 | 否 |
| Phone | varchar | 20 |  |  | 电话 |  |
| Contact | varchar | 30 |  |  | 联系人 |  |
| Status | Int |  |  |  | 状态，是否取消，0 未处理，1已处理，2 取消 |  |
| Remark | Varchar | 1024 |  |  | 备注 |  |

（29）维修表，该表用来记录维修信息。

表5-27 维修表（mms\_maintain）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| id | int | 11 | NOT NULL | 是 | 主键 | 是 |
| Staff | varchar | 1024 | NOT NULL |  | 维修员工 | 否 |
| AppointmentId | int | 11 | NOT NULL |  | 预约id | 否 |
| CreateTime | Date | 20 | NOT NULL |  | 创建时间 | 否 |
| Status | int | 255 | NOT NULL |  | 状态 | 否 |
| Type | varchar | 30 | NOT NULL |  | 车型 | 否 |
| OCU | Varchar | 30 | NULL |  | 创建账号 | 否 |
| OCD | Date |  | NULL |  | 创建时间 | 否 |
| LUC | Varchar | 30 | NULL |  | 最后更新账号 | 否 |
| LUD | Date |  | NULL |  | 最后更新时间 | 否 |
| RETURN\_TIME | datetime |  |  |  | 交付时间 |  |
| MAINTAIN\_NO |  |  |  |  | 表单编号 |  |

（30）旧配件表，该表用来登记维修车辆更换下来的旧零件。

表5-27 旧配件表（mms\_maintain\_oldpart）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| id | int | 11 | NOT NULL | 是 | 主键 | 是 |
| MaintainId | Int | 20 | NOT NULL |  | 维修单 | 否 |
| Type | varchar | 255 | NOT NULL |  | 规格 | 否 |
| Brand | varchar | 30 | NOT NULL |  | 品牌 | 否 |
| Number | int | 30 | NULL |  | 数量 | 否 |
| Price | Decimal |  | NULL |  | 价格 | 否 |
| Status | Varchar | 50 | NULL |  | 状态 | 否 |
| Remark | Varchar | 255 | NULL |  | 备注 | 否 |

（32）维修工具表，该表用来登记维修车辆用到的零件。

表5-27 维修工具表（mms\_miantain\_tools）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **宽度** | **允许空值** | **主键** | **说明** | **是否唯一** |
| id | int | 11 | NOT NULL | 是 | 主键 | 是 |
| MaintainId | Int | 20 | NOT NULL |  | 维修单 | 否 |
| SkuId | varchar | 255 | NOT NULL |  | 规格 | 否 |
| Number | int | 30 | NULL |  | 数量 | 否 |
| Status | Int |  | NULL |  | 是否已经归还，1已经归还，0未归还，2丢失 | 否 |
| Remark | Varchar | 255 | NULL |  | 备注 | 否 |
| Price | Decimal |  | NULL |  | 价格 | 否 |
| Damage | Decimal |  |  |  | 赔偿金额 | 否 |

**接口文档**

## **约定**

服务器基址：（待定）

#### 请求数据返回

{

"code": 101 || 102 || 202 || 303,

"message": "error message",

"extend": {} || [] || ""

}

说明：

code:

101：成功；102：访问成功，但条件不足，过程并没有出错；202：服务器异常，出错了；303：权限不足；

message:

错误信息；

extend:

成功访问后的返回值。

**用户模块**

1. 注册

路径：/passport/register

请求方式：Post

请求数据：（qs.stringfy）

{

UserName: ‘’,

Pwd: ‘’

}

返回数据：

{

"code": 101 || 202, // 成功 || 失败 （后面的同理）

"message": "error message", // 成功，message为空字符串

//（后面的同理）

"extend": ""

}

1. 登录

路径：/passport/login

请求方式：post

请求数据：

{

UserName: “”,

Pwd: “”

}

返回数据：

{

“code”: 101 || 102 || 202,

“message”: “error message” || “no condition message”, // 当code === 102 时，条件不 // 足信息也在这里（后面的同理）

“extend”:

{

“id”: 1,

“UserName”: “”,

“Email”: “”,

“Phone”: “”,

“RoleList”: [

{

“id”: 1,

“RoleName”: “”,

“Description”: “”,

“AuthorityList”: [

{

“id”: 1,

“AuthorityName”: “”,

“Description”: “”,

“Icon”: “”, // 菜单图标

“Resource”: “”, // 资源路径

“ParentId”: 0, // 父节点id，一级菜单为0

“Level”: 1, // 菜单级别

}

]

}

],

“AuthorityList”: [ // 提取出来方便菜单处理

{

“id”: 1,

“AuthorityName”: “”,

“Description”: “”,

“Icon”: “”, // 菜单图标

“Resource”: “”, // 资源路径

“ParentId”: 0, // 父节点id，一级菜单为0

“Level”: 1, // 菜单级别, 从1级开始

}

]

}

}

1. 权限管理

3.1 添加权限

路径：/authority/add

请求方式：post

请求数据：

{

“AuthorityName”: “”,

“Description”: “”,

“Icon”: “”,

“Resource”: “”,

“ParentId”: 0,

“Level”: 1,

}

返回数据

{

“code”: 101 || 202 || 303,

“message”: “”,

“extend”:

{

“id”: 1,

“AuthorityName”: “”,

“Description”: “”,

“Icon”: “”,

“Resource”: “”,

“ParentId”: 0,

“Level”: 1,

}

}

3.2 更新权限

路径：/authority/update

请求方式：post

请求数据：

{

“id”: 1,

“AuthorityName”: “”,

“Description”: “”,

“Icon”: “”,

“Resource”: “”,

“ParentId”: 0,

“Level”: 1,

}

返回数据

{

“code”: 101 || 202 || 303,

“message”: “”,

“extend”: ""

}

3.3 删除权限

路径：/authority/delete

请求方式：delete

请求数据：

{

“id”: 1,

}

返回数据

{

“code”: 101 || 202 || 303,

“message”: “”,

“extend”: ""

}

3.4 查询权限

查询所有

路径：/authority/getAll

请求方式：get

请求数据：

返回数据

{

“code”: 101 || 202 || 303,

“message”: “”,

“extend”: [

{

“id”: 1,

“AuthorityName”: “”,

“Description”: “”,

“Icon”: “”,

“Resource”: “”,

“ParentId”: 0,

“Level”: 1,

}

]

}

条件查询

路径：/authority/search

请求方式：get

请求数据：

{

“SerachStr”: “”

}

返回数据

{

“code”: 101 || 202 || 303,

“message”: “”,

“extend”: [

{

“id”: 1,

“AuthorityName”: “”,

“Description”: “”,

“Icon”: “”,

“Resource”: “”,

“ParentId”: 0,

“Level”: 1,

}

]

}

查询一级权限

路径：/authority/getLevel1

请求方式：get

请求数据：

返回数据

{

“code”: 101 || 202 || 303,

“message”: “”,

“extend”: [

{

“id”: 1,

“AuthorityName”: “”,

“Description”: “”,

“Icon”: “”,

“Resource”: “”,

“ParentId”: 0,

“Level”: 1,

}

]

}

查询二级权限

路径：/authority/getLevel2

请求方式：get

请求数据：

{

“ParentId”: 1

}

返回数据

{

“code”: 101 || 202 || 303,

“message”: “”,

“extend”: [

{

“id”: 1,

“AuthorityName”: “”,

“Description”: “”,

“Icon”: “”,

“Resource”: “”,

“ParentId”: 0,

“Level”: 2,

}

]

}

4 角色管理

4.1 添加角色

路径：/role/add

请求方式：post

请求数据：

{

“RoleName”: “”,

“Description”: “”,

“AuthorityListStr”: “1-2-3-4”, // 用 ‘-’ 拼接权限id 字符串

}

返回数据

{

“code”: 101 || 202 || 303,

“message”: “”,

“extend”:

{

“id”: 1,

}

}

3.2 更新角色

路径：/role/update

请求方式：post

请求数据：

{

“id”: 1,

“RoleName”: “”,

“Description”: “”,

“AuthorityListStr”: “1-2-3-4”, // 用 ‘-’ 拼接权限id 字符串

}

返回数据

{

“code”: 101 || 202 || 303,

“message”: “”,

“extend”: ""

}

3.3 删除权限

路径：/role/delete

请求方式：delete

请求数据：

{

“id”: 1,

}

返回数据

{

“code”: 101 || 202 || 303,

“message”: “”,

“extend”: ""

}

查询角色

1. 公司/客户管理

3.1 客户添加

路径：/client/add

请求方式：post

请求数据：

{

“Company”: “”,

“Description”: “”,

}

返回数据：

{

“code”: 101 || 102 || 202 || 303,

“message”: “error message” || “exists”,

“extend”: “”

}

3.2 客户修改

路径：/client/update

请求方式：post

请求数据：

{

id: 1,

“Company”: “”,

“Description”: “”,

}

返回数据：

{

“code”: 101 || 102 || 202 || 303,

“message”: “error message” || “exists”,

“extend”: “”

}

3.3 客户删除

路径：/client/update/id

请求方式：post

请求数据：

返回数据：

{

“code”: 101 || 102 || 202 || 303,

“message”: “error message” || “exists”,

“extend”: “”

}

3.4 查询所有

路径：/client/getAll

请求方式：get

请求数据：

返回数据：

{

“code”: 101 || 202 ,

“message”: “error message” ,

“extend”: [

{

“id”: 1,

“Company”: “”,

“Description”: “”

}

]

}

3.5 条件查询

路径：/client/get

请求方式：get

请求数据：

{

“Company”: “”

}

返回数据：

{

“code”: 101 || 202 ,

“message”: “error message” ,

“extend”: [

{

“id”: 1,

“Company”: “”,

“Description”: “”

}

]

}

4 预约管理

4.1 预约添加

路径：/appointment/add

请求方式：post

请求数据：

{

“CompanyId”: 1,

“Description”: “”,

“”

}

返回数据：

{

“code”: 101 || 102 || 202 || 303,

“message”: “error message” || “exists”,

“extend”: “”

}

SqlConnectionFactory

"uid=sa;pwd=GrandPower.tech;initial catalog=IFTM\_HOTEL;data source=192.168.1.10;Connect Timeout=2000"