|  |  |
| --- | --- |
| **保密等级** | **内部** |
| **文档名称** | **文档模板** |
| **文档编号** | **GMWL-NJ-D-001** |
| **发布组织** | **研发部** |
| **发布日期** | **2018年5 月 日** |
| **执行日期** | **2018年 月 日** |
| **版 本 号** | **V.1.0** |

**中通国脉物联科技**

**基础开发框架说明**

|  |  |  |
| --- | --- | --- |
| **批准人签字** | **审核人签字** | **制订人签字**  **徐仕强** |
| **日期： 2018/ /** | **日期：2018/ /** | **日期：2018/7/** |

**中通国脉物联科技南京有限公司**

**变更履历**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **版本编号** | **简要说明(变更内容、变更位置、变更原因和变更范围)** | **变更日期** | **变更人** | **审核人** | **批准人** | **批准日期** |
| 1 | V.1.0 | 创建 | 2019-1-21 | 张杰 |  |  |  |

目录

[1. 前言 4](#_Toc26509)

[2. 说明 4](#_Toc9012)

[2.1. 基础表结构 4](#_Toc11312)

[2.2. 项目基础说明 5](#_Toc30294)

[2.2.1. 目录 5](#_Toc24002)

[2.2.2. 表的命名 5](#_Toc13910)

[2.2.3. 表的备注 5](#_Toc7875)

[3. 功能 6](#_Toc29426)

[3.1. 权限管理 6](#_Toc25136)

[3.1.1. 菜单权限管理 6](#_Toc21640)

[3.1.2. 角色管理 6](#_Toc29433)

[3.1.3. 用户管理 7](#_Toc22054)

[3.2. 自动生成管理 7](#_Toc21585)

[3.2.1. 模板说明 7](#_Toc31015)

[3.2.2. 配置说明 8](#_Toc10576)

[3.2.3. 界面及使用说明 9](#_Toc1737)

# 前言

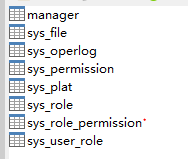
本框架目标为方便后续系统的快速开发，

以IOT平台框架为基础，整合的 springboot基础开发框架.

# 说明

## 基础表结构





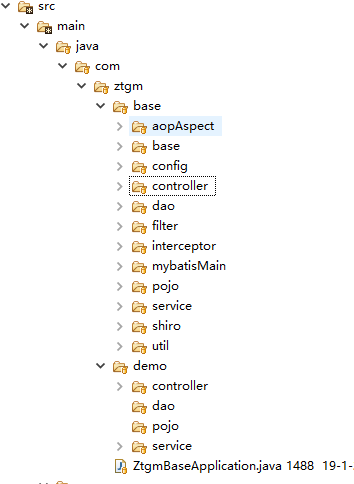
包含权限/菜单/用户/平台/上传附件/系统日志 基础最小框架.

## 项目基础说明

### 目录

Com/ztgm/base目录为框架基础功能（包含权限等）

Com/ztgm/demo （demo为示例，可根据不同项目修改），目录为自动生成的代码放置位置.(可配置)



### 表的命名

框架自动生成规则为 数据表的命名需要下划线分割， 如：

表：fee\_user\_type，

自动生成后生成的Pojo 为 FeeUserType （驼峰模式.）

### 表的备注

框架自动生成页面增删改查过程使用了表结构中的备注字段作为页面默认显示字段.

表/及表中每个字段最好都补上备注说明

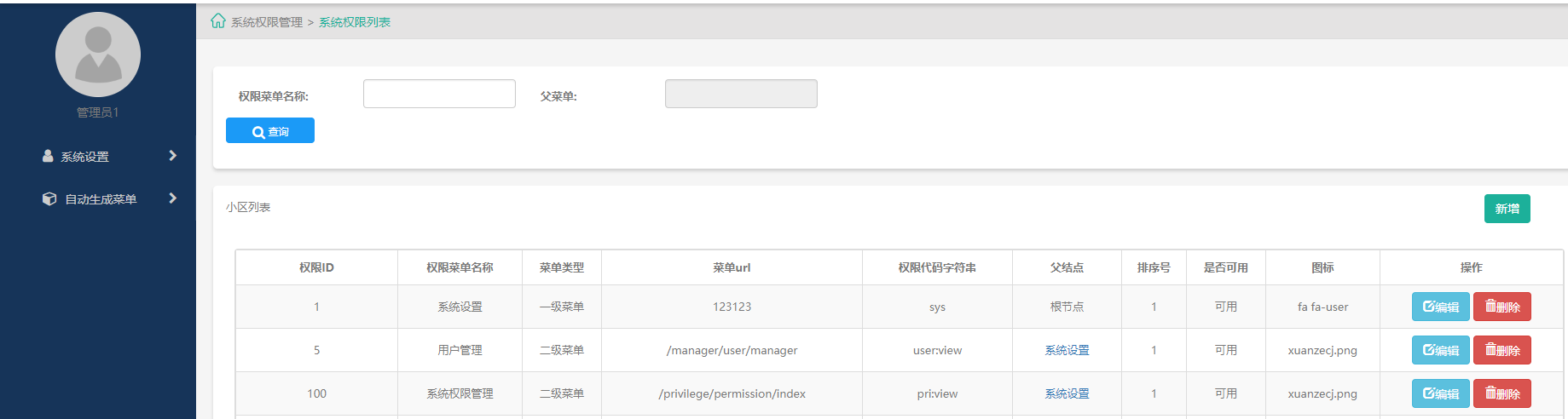
# 功能

框架包含功能

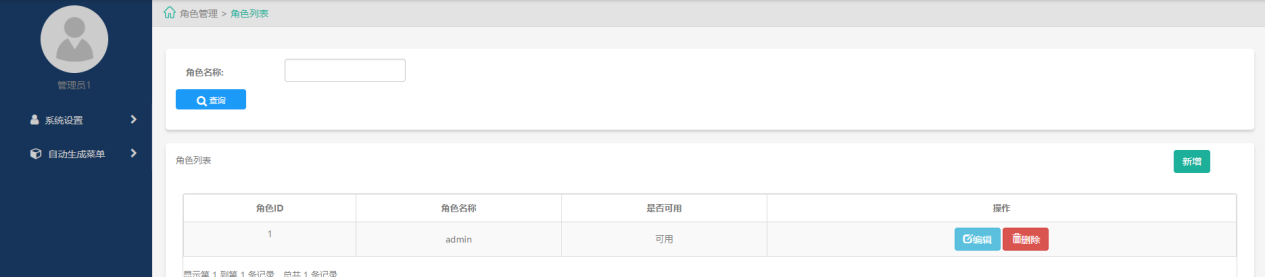
* 权限管理
* 基础单表的增删查改代码生成功能。
* 文件上传前后台相关基础组件.
* 左侧菜单优化为二级弹出式样式.

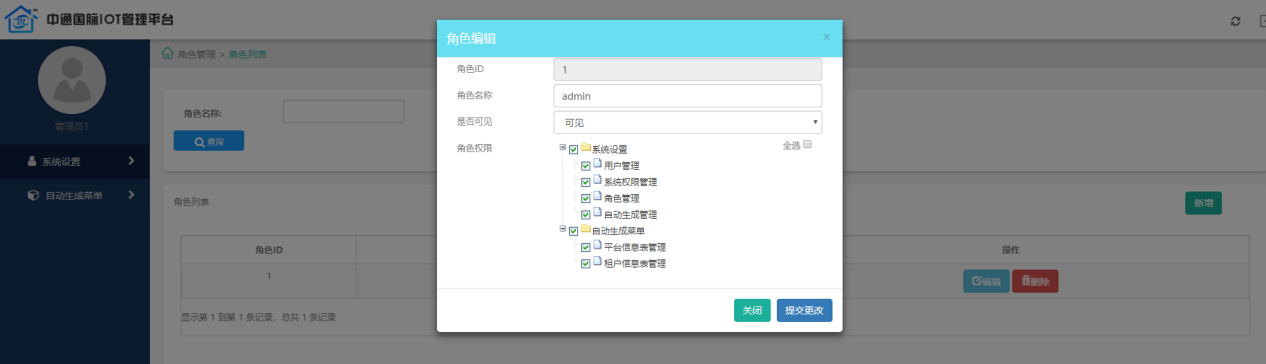
## 权限管理

### 菜单权限管理

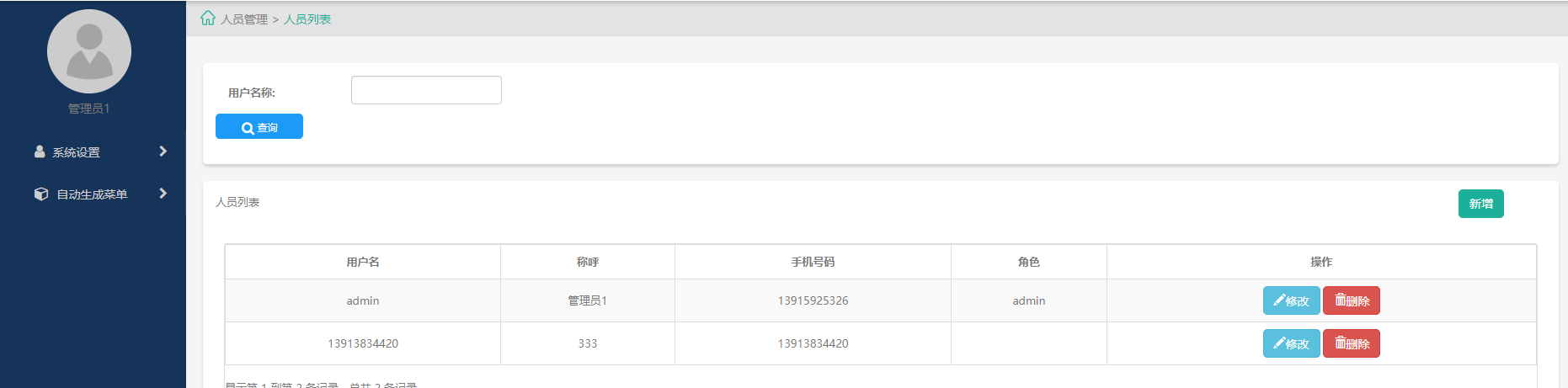


### 角色管理





### 用户管理



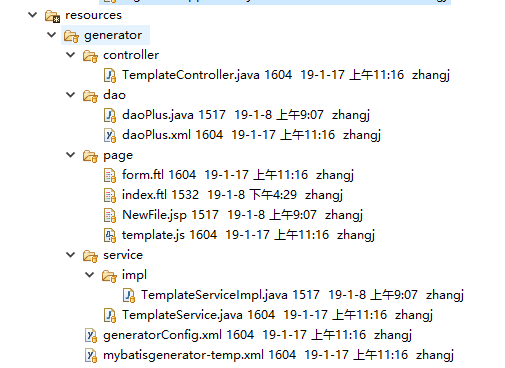


## 自动生成管理

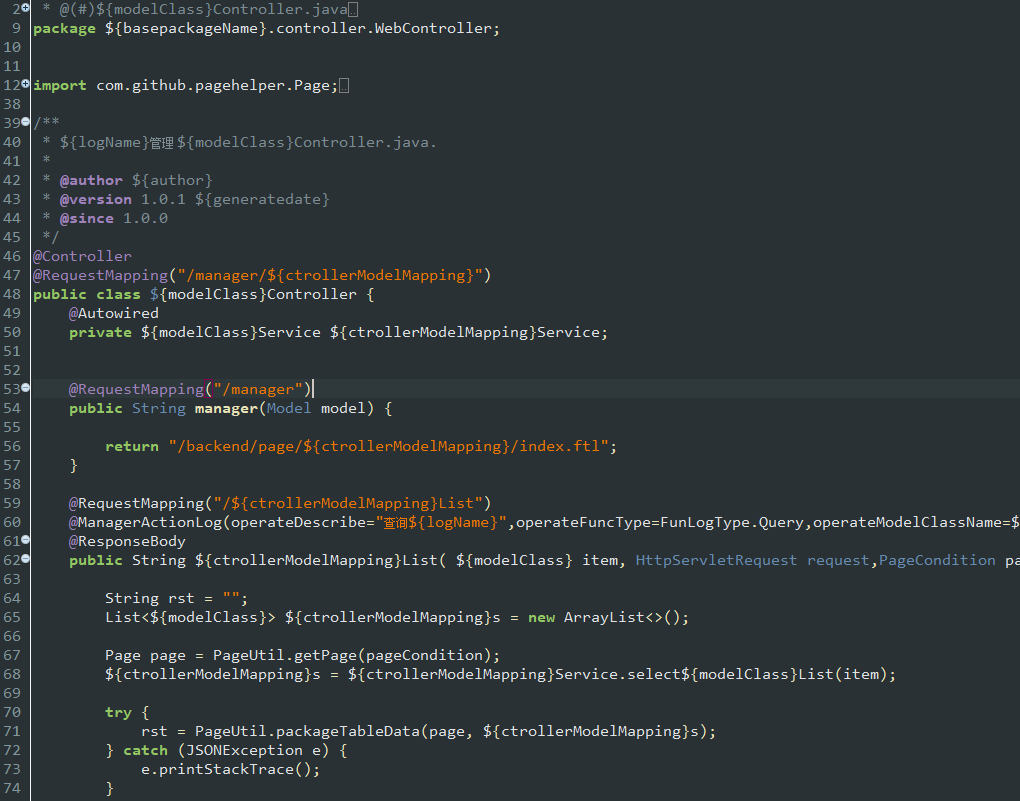
整体基于Freemarker，使用预定义模板生成相关java及js,ftl,jsp(待扩展)文件.

### 模板说明

整体模板位于resources下的generator目录。分为controller/dao/page/service类型



全部模板文件均基于Freemarker语法格式，根据前台界面选择的相关参数实现模板的生成。



Pojo/Dao的生成方式如下：

1.调用了MyBatisGenerator 生成基础

2.然后对生成的文件补充了自定义查询等相关方法.

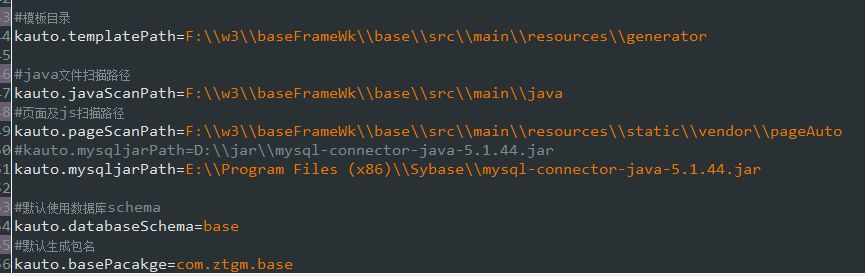
### 配置说明

相关基础配置参数位于 resources/project.properties文件中。

本地开发时需要针对本地情况修改各自的配置文件。

Kauto.databaseSchema 查询的库

kauto.basePackage 默认生成相关文件的包名



### 界面及使用说明

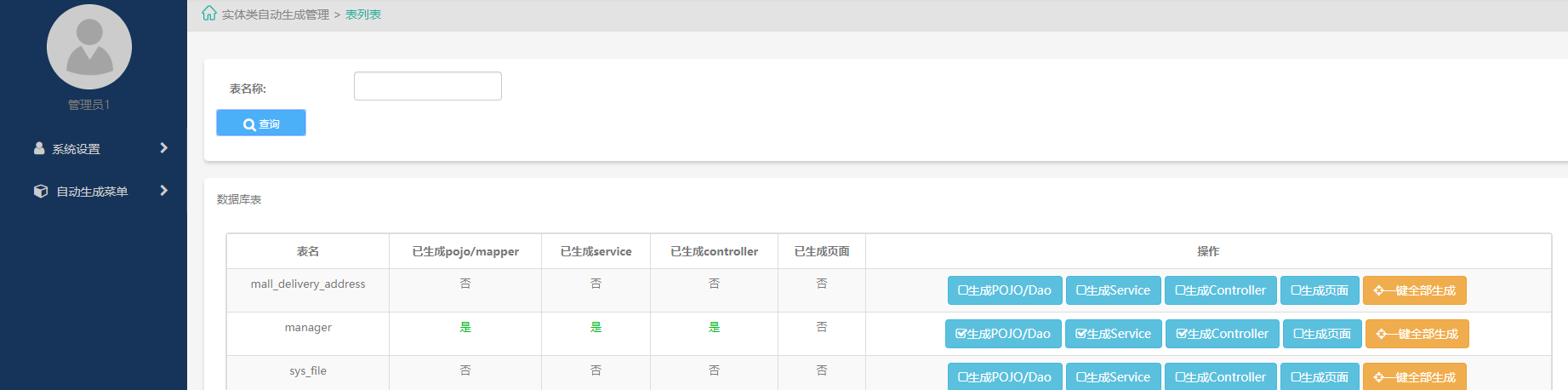
可以选择生成单独的Pojo/Dao, Service,Controller,页面

或者一键全部生成。

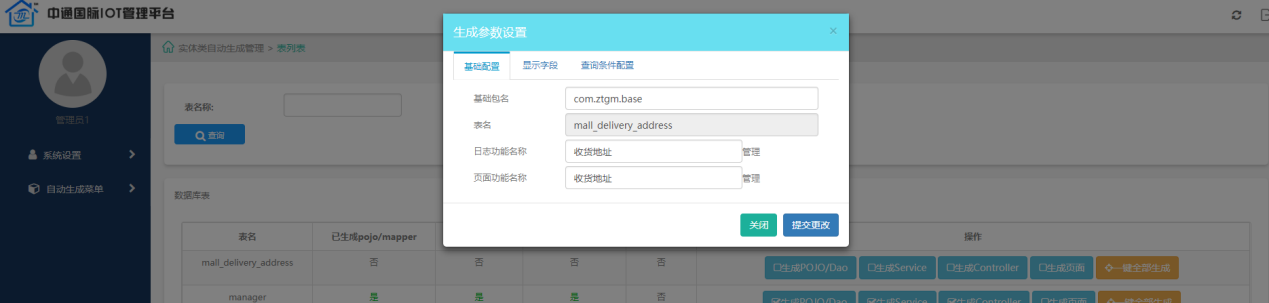
生成过程除 Dao对应的XXmapper.xml文件会备份一个放到resource/temp目录外，

其他文件生成过程会覆盖已有相同包下面的代码。

生成时，页面会提示覆盖确认。



基础参数



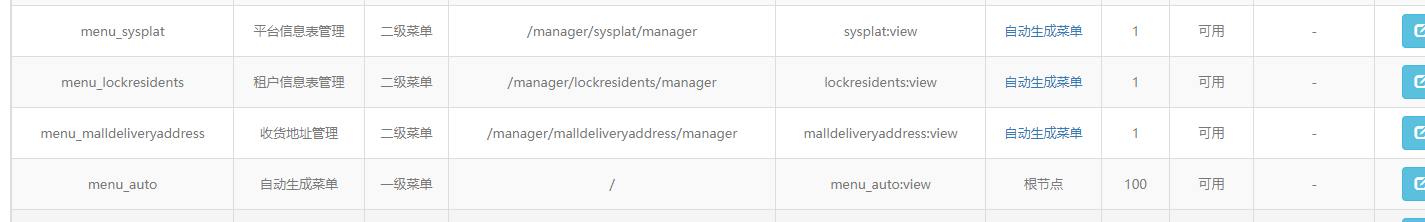
列表字段，根据数据库表中的字段备注默认生成，可在页面修改



基础查询条件，默认选择前两个字段.



生成完成会自动在当前用户第一个角色下新建一个对应的菜单项。



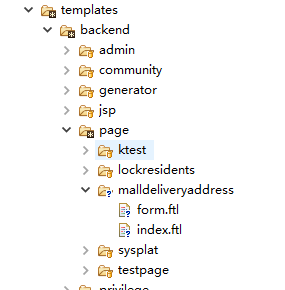
生成完成，需要重启本地项目，编译加载生成的java文件，若只是调整页面查询条件等，可以不用重启。

使用页面右上角的刷新可以重新加载菜单及页面.

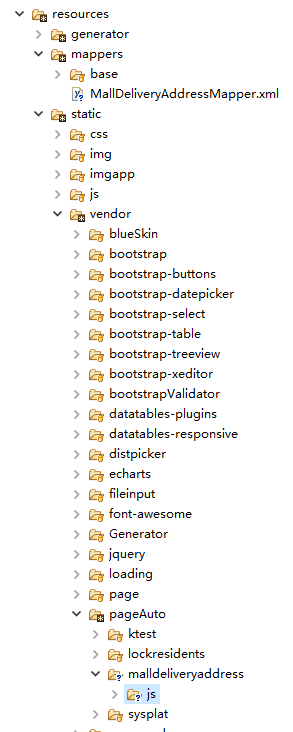


生成的菜单默认放置在自动生成菜单中，可以后续自己配置至其他目录。

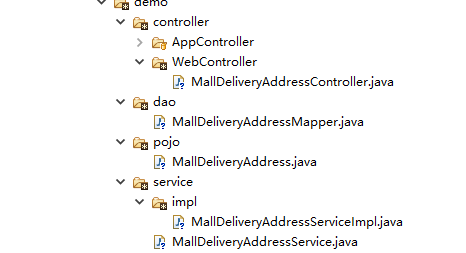
生成的页面在 templates/backend/page/下。



对应的js在static/vendor/pageAuto下



Java代码在页面指定的Pacakge目录。

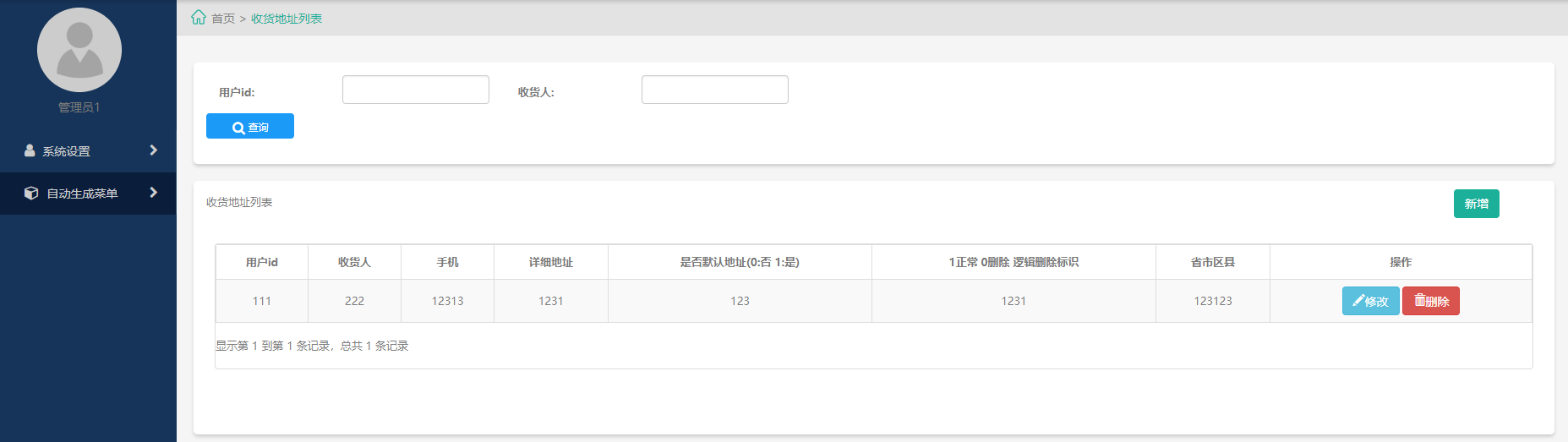


重启加载项目

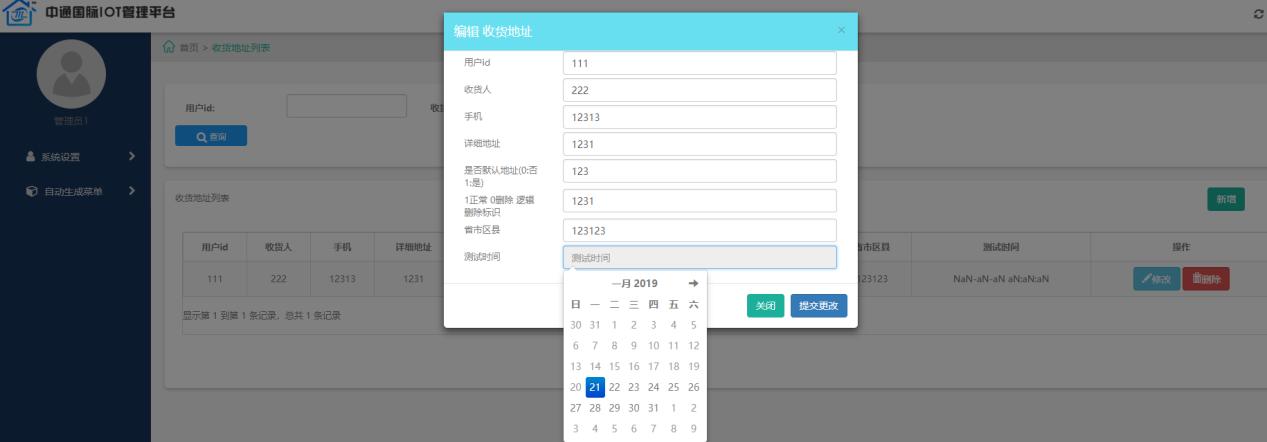


默认生成的页面已经具备全部增删查改功能。





数据表中为timestamp的字段，页面上默认使用 bootstap 的datetimepicker组件实现。



## 文件上传

方案同商城项目，文件通过本地上传至本地后台后，后台会再次转发文件，将图片上传至指定的文件服务器（可以使用本地项目/或者远程部署的后台）集中存放.

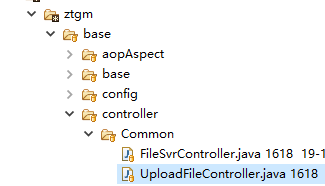
文件服务器返回文件的md5编码并记录在文件服务器（本地或远程）后台库中.

项目中需要图片的表中增加文件md5记录字段，通过文件md5请求后，后台重定向至文件服务器通过md5获取文件.

### 后台封装

UploadFileController本地文件上传

FileSvrController 文件服务器实现。



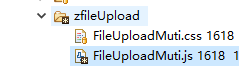
### 配置

配置位于resources/project.properties中。



### 前台上传组件

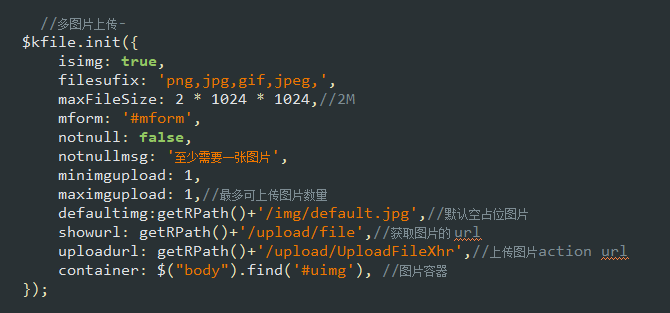
Resources/stastic/vendor



封装图片上传组件，支持上传图片或者上传文件/支持最小/最大上传数量/上传文件后缀，大小限制等.具体使用参考用户管理图片修改.

使用：

初始化



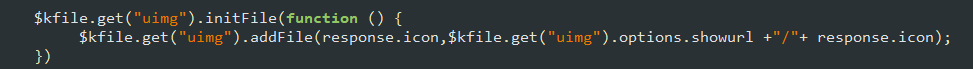
空图片



重置验证

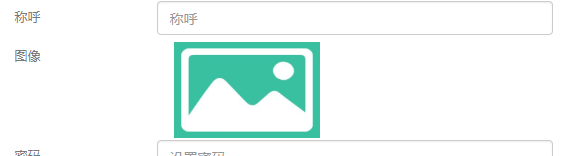


修改时，默认加载已有图片

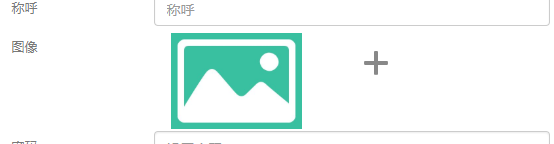


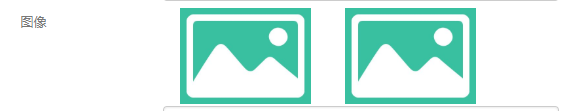
界面：

单图片



多图片





多文件



点击图片上传:

