

# 前言：

---

每天的结构就是：

//默认情况所谓遇到的问题是指已解决，如果有未解决，就再加一个枝干

// 如果有笔记那就笔记

.

├做了

├计划

├遇到的问题

└笔记

.

├做了

├计划

├遇到的问题

├遇到的问题（未完成）

└笔记

## 第一周

---

### 2023.12.05

- 做了：
  1. 下载配置大部分可能需要的开发工具、及其相关工具还有自己常用工具，git巴拉巴拉，基本都是网络或现有文件，或现有资料下载的。
  2. 阅读并熟悉java开发规范（阿里巴巴版本）
  3. 在集成开发环境查看下下来的cloudx前后端项目
- 计划：
  - 熟悉业务
- 遇到的问题：
  - 因为我那公司的电脑昨天下午才到，然后网络还只能访问内网，所以自己搞网络花了点时间，今早才找主管帮忙，但是今天也没能把那网络整好。（来自第二天的吐槽：莫名其妙的有可以上外网了，玄学）
  - 我这一天半里大部分时间是在安装和配置工具，然后还遇到了报错，那个报错花了我一小时多左右，比如nacos启动时的jdk路径有空格导致启动不起来，以及mysql启动需要vsc++环境。然后也因为没有网络的原因，所以整体速度会慢一些
  - 期间也有大致看公司的规范；然后还去看了关于go的内容，看go看了很久，看go基础，相关项目，并记录笔记。

### 2023.12.06

- 做了：
  1. 上午下载Chrome、加速器、配置并登入常用网站、下载调节亮度的插件、测试typora的上传图片功能、想实现右键打开（但是没实现）
  2. 找壁纸，好用的壁纸网站"[Download this awesome wallpaper - Wallpaper Cave](#)
  3. 配置电脑亮度
  4. 启动前后端项目(前端:cloudx-ui；后端：estate微服务)，并大致查看

- 计划：
  - 看estate项目（补充：[idea快速生成结构图](#)）
  - 结合开发规范看文档
- 遇到的问题：
  - 报错：Command line is too long. Shorten command line。解决方式：[IDEA命令行缩短器助你解决此问题：Command line is too long. Shorten command line-CSDN博客](#)；[idea运行XXX时出错。命令行过长。通过JAR清单或通过类路径文件缩短命令行，然后重新运行。-CSDN博客](#)
  - nodejs文件结构问题导致的安装的包找不到命令
  - npm里saas问题(反正上面的解决了.报错就少了很多,然后上网搜就好了.我已经找不到原链接了,历史记录也没有，总之就是让npm config registry set "sass\_binary\_site" = "<https://npm.taobao.org/mirrors/node-sass>")
  - host修改问题
  - nginx代理问题（nginx.conf配置代理端口）
  - nginx启动问题（配置好nginx文件目录）
  - 前端缓存问题（ctrl+5）
  - 可能是idea缓存的问题，导致项目不正常启动，就是启动的端口不是指定端口应该（玄学）
- 遇到的问题(未解决):
  - ~~想实现右键打开idea（但是没实现）~~

## 2023.12.07

- 做了
  1. 熟悉前端
  2. 配置idea主题和notepad++主题
  3. 看estate项目（补充：[idea快速生成结构图](#)），通过前端页面大致看了项目的界面和功能
  4. 细看了登入页鉴权、首页的功能及其对应的后端代码
  5. 大致看了房屋管理的业务
- 计划
  - 看主要业务的逻辑
  - 看第三方腾讯云对讲（后面看）
- 遇到的问题
  - 我该侧重什么？业务？代码？应该是逻辑（住户的还有设备还有房屋的）代码要搞懂还是只需要知道这个的功能就好了？答：暂时只需要功能，主要是把全部业务熟悉一下。那我需要
- 遇到的问题(未解决)
  - 鉴权上面的注释有必要么
- 笔记：

### 实用技巧：

- restful根据前端请求地址快速定位接口位置(ctrl+alt+/,)
- 前端ctrl+shift+f快速找到异步请求位置
- 后端ctrl+shift+f快速找到需要的文件
- get用于查询post为非查询

## 控制类新参及函数接口常见参数：

- json格式就用@requestBody
- 基本类型或集合类型就用@RequestParam
- 日期用@DateTimeFormat(这一般是放对象里配置的,相当与实体接收参数)
- 路径用@PathVariable
- 数组就是数组名和参数名保持一致,基本类型可以参数名保持一致
- [@RequestBody和@RequestParam和@RequestPart使用场景 Arrowyy的博客-CSDN博客](#)
- [swagger2 注解说明 @ApiImplicitParam和@ApiImplicitParams的用法](#)
- [api注解](#)

补充：

- [\(2封私信 / 18 条消息\) npm 和 yarn 那个好用? - 知乎\(zhihu.com\)](#)
- **tudo**: 前端传集合参数乱码myajax

## 首页未登入时的鉴权：

```
@ApiOperation(value = "获取授权信息并且校验")
@GetMapping("/getSysInfo")
@ApiImplicitParams({
    @ApiImplicitParam(name = "id", value = " 集团id 项目组id 或 项目id",
        paramType = "path", required = true),
    @ApiImplicitParam(name = "Authorization", value = "授权token",
        required = true, paramType = "header"),
    @ApiImplicitParam(name = "TENANT-ID", value = "租户Id", required =
        true, paramType = "header"),
    @ApiImplicitParam(name = "PROJECT-ID", value = "小区Id", required =
        true, paramType = "header")
})
@Inner(false)
public R getSysInfo() {
    // 获取当前激活的应用程序配置文件
    if("dev".equals(SpringContextUtil.getProfile())) {
        return R.ok();
    }
    // 从Redis中获取预先存储的授权信息
    String systemInfo = (String) RedisUtil.get("systemInfo");
    if(null == systemInfo) {
        // 获取所有授权信息
        return sysInfoService.verifiedAll();
    }
    // 将上面获取所有授权信息, 存储到Redis中
    return R.ok(RedisUtil.get("systemInfo"));
}
```

```
spring:
profiles:
  # 在 Spring 的配置文件中，我们可以通过 spring.profiles.active 属性来指定当前激活
  的配置文件。在配置中，spring.profiles.active 的值为 '@profiles.active@'，表示使用
  @profiles.active@ 占位符来引用激活的配置文件名。
  # 在 Spring 启动时，它会根据激活的配置文件来加载相应的配置。例如，如果当前激活的配
  置文件是 dev，那么 spring.profiles.active 的值将被替换为 dev，从而加载 application-
  dev.yml 文件中的配置。
  # 使用占位符可以方便地实现配置的动态切换，而不需要手动修改配置文件。当需要切换到其他配
  置文件时，只需修改 spring.profiles.active 的值即可。例如，将 active 属性设置为 prod
  将会加载 application-prod.yml 文件中的配置。
  active: '@profiles.active@'
```

补充: [spring.profiles.active=@profiles.active@ 如何使用idea讲-CSDN博客](#)

## setup语法糖

[Vue3 中setup\(\)和 <script setup> </script> cannot contain es module exports-CSDN博客](#)

[vue3 setup语法糖请注意在script标签里加setup, 不用return在script 标签上直接写setup有两把刷子的博客-CSDN博客](#)

[一文看懂vue3中setup\(\)和 <script setup> <script> 的区别的setup\(\)的区别努力学前端的小F的博客-CSDN博客](#)

## int4和int8区别

MySQL中的整数类型主要有两种:

int4: 它用于存储需要较小范围的整数, 例如年龄、金额等。它可以存储从-2147483648到2147483647之间的整数, 并且通过查询和排序等操作可以有效地提高数据库的查询效率。

int8: 它用于存储需要存储大量数据或需要更高精度的计算场景, 例如时间戳和唯一标识符等。它可以存储从-9223372036854775808到9223372036854775807之间的整数。

## 2023.12.08

- 做了:
  1. 同步常用笔记 (上传所有本地图片报错想解决但是入迷了, 也没解决)
  2. 购物: 充电器+鼠标 (13.00-14.30)
  3. 记账: 近期账本的全部 (14.30-15.30)
  4. 看人工智能文章: [captainbed.cn](#) (但是现在是666,明天就是1024, 等我哪天不计较这个钱的时候我再买, 毕竟我只是爱好者)
  5. 摸鱼的一天(^ - ^)V
- 计划:
  - 看主要业务的逻辑
- 遇到的问题:
- 遇到的问题(未解决):
  - piggo的nodejs有问题:

```
2023-12-08 11:27:38 [PicGo ERROR] {
  "method": "PUT",
  "url":
  "https://api.github.com/repos/nmdsbzmhbky/images/contents/imgs/20231208
  1127193.png",
  "statusCode": 0,
  "message": "socket hang up",
  "stack": "Error: socket hang up\n    at connResetException
(node:internal/errors:691:14)\n    at TLSSocket.socketOnEnd
(node:_http_client:471:23)\n    at TLSSocket.emit (node:events:406:35)\n
    at endReadableNT (node:internal/streams/readable:1343:12)\n    at
processTicksAndRejections (node:internal/process/task_queues:83:21)",
  "response": {
    "status": 0,
    "statusCode": 0,
    "body": ""
  }
}
```

- 笔记:

## 第二周

2023.12.11

- 做了
  1. 因为表太多了，找了MysqlWorkBench，安装并配置vc以及远程连接。主要是适合看er图
  2. [推荐给你3个完美替代 Navicat 的工具 \(qq.com\)](#)
  3. [MySQL Workbench连接mysql - KAZUYA - 博客园 \(cnblogs.com\)](#)
  4. 细看住户登记下的住户信息数据库
  5. idea文件设置: [idea设置选中文件时，左侧自动跳转到文件所在位置 idea设置打开文件左侧选中-CSDN博客](#)
- 计划
- 遇到的问题
  - 安装MysqlWorkBench报错-MYSQL workbench requires the Visual C++ 2019 Redistributable Package to be .....: [MySQL Workbench安装MySQL Workbench requires the Visual C++ 2019 Redistributable Package to be .....](#)
- 笔记
  - VO, BO, PO, DO, DTO的关系: [一篇文章讲清楚VO, BO, PO, DO, DTO的区别](#)
  - 递归查询with recursive的使用【[MYSQL WITH recursive使用](#)】 [mysql的recursive](#)
  - group\_concat函数连接[MySQL中group\\_concat函数用法总结 - 古兰精 - 博客园 \(cnblogs.com\)](#), 注意: 此时如果要排序, group必须放在函数中间。

- 逻辑顺序：  
`from > where > group by > having > order by > select`  
 选表      选行      对表      对组筛选      对组/表排序

在`select`的字段前面就是对前面得出的每行进行操作（但是如果有`group by`，出现了字段，那所对应的字段就是一个组的，但是有时又要看用的是什么函数/逻辑程序，如果是函数，那就是括号里的，逻辑就是前者）

- [idea设置选中文件时，左侧自动跳转到文件所在位置 idea设置打开文件左侧选中-CSDN博客](#)

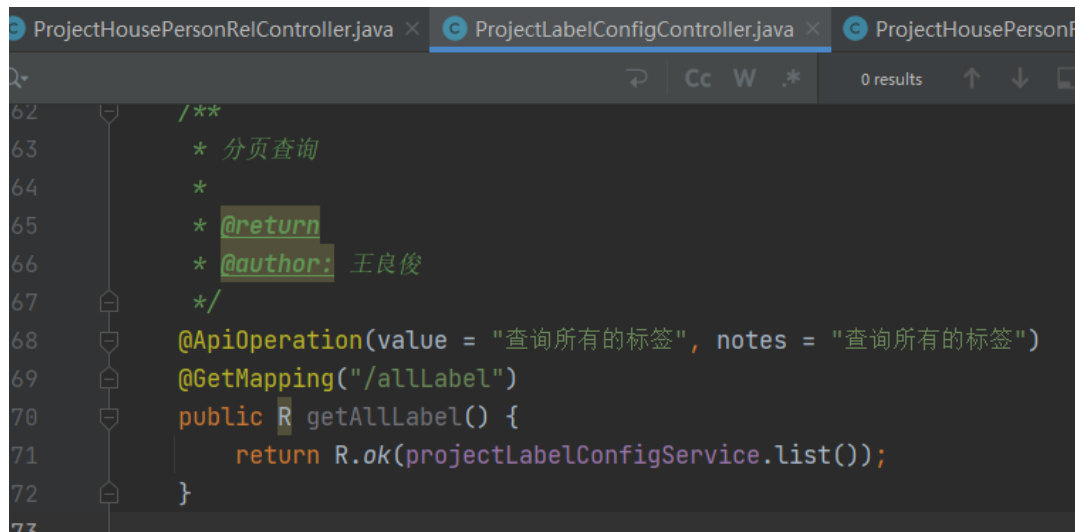
## 2023.12.12

### • 做了

1. 住户登记下的新增功能
2. 查询所有标签
3. 下了2021.2.2版本idea，之前的版本是直接用过公司的2020的好像，即使破解了，也还是缺少功能，比如database没有，然后没法配置test文件，右键打开有问题。然后现在这些问题都解决了！
4. 熟悉database的用法，如：sql的反向定位数据库表、console的使用、数据表界面的使用、输出表界面的使用
5. 细看 查询楼宇基本信息 接口

### • 遇到的问题未解决)

1. console界面快速找到字段无法实现
2. `/estate/baseLabelConfig/allLabel`



```

62  /**
63   * 分页查询
64   *
65   * @return
66   * @author: 王良俊
67   */
68  @ApiOperation(value = "查询所有的标签", notes = "查询所有的标签")
69  @GetMapping("/allLabel")
70  public R getAllLabel() {
71      return R.ok(projectLabelConfigService.list());
72  }
73
  
```

3. `/estate/projectPersonAttr/1000000708/1/undefined`

### • 笔记:

- [idea技巧之sql反向定位数据库表，以及智能提示杨虎成的技术博客51CTO博客](#)

2023.12.13

- 做了

1. 给查询楼宇基本信息接口控制类注释
2. 修改restfultool搜索接口的快捷键，修改方式：keymap->pluageinx->restfultool
3. 测试类中循环list: [idea中各种for循环的快捷键](#) [idea 快速写循环list-CSDN博客](#)
4. 编写并debug测试类：

- 先说小结：

前言：引入Junit和springstartertest依赖

首先：解决了命令行太长在run/debug config以及.idea下的.xml文件进行配置

其次：对package进行修改并对test文件配置为Test Resource Root

然后：将test和java包包名进行相对应

最后：在test类上面标注@RunWith(SpringRunner.class)

@SpringBootTest(webEnvironment =

SpringBootTest.WebEnvironment.RANDOM\_PORT)，成功可以使用test

啦！！！！！！！！！！

- Command line is too long: [Idea单测执行报错“Command line is too long”解决办法](#)  
[AND](#) [IDEA-解决Command line is too long. Shorten.....](#)
- test文件红色感叹号，package错误修改: [idea @Test报错，带有红色感叹号 - 简书](#)  
([jianshu.com](#))
- Unable to find a @SpringBootConfiguration, you need to use  
@ContextConfiguration: [SpringBoot测试失败并报错: Unable .....](#)
- Failed to load ApplicationContext: [解决: Failed to load ApplicationContext-CSDN](#)  
[博客](#)

但是还是报这个错，然后由这篇文章为灵感[关于Failed to load ApplicationContext](#)  
[报错的原因](#) [AND](#)找到根本上的错误[SpringBoot启动单元测试报错](#)  
[javax.websocket.server.ServerContainer not available](#)

5. 看接口：

- a. estate/baseLabelConfig/allLabel：查询项目下的标签(涉及到mp的多租户插件,用来对sql语句进行额外的处理的，即多租户分页插件)[Mybatis-plus多租户](#)和[插件扩展 - 多租户SQL 解析器](#)补充:[filter方法中返回true](#)
- b. estate/projectPersonAttr/{projectId}/{type}/{personId}：查询拓展人员属性的接口(看不懂)
- c. estate/baseBuildingFrame/listPuid/{puid}：通过puid获取（楼宇）框架信息接口
- d. estate/baseHousePersonRel/checkHasPreRegister：房屋预注册检查接口（需要结合电话号码检查，因为有的房屋可以有多个住户，所以必须一个住户对某间房屋的持有只有一份数据！！前端最好在电话和房屋都存在的情况下，在向后端发送请求,只是需要用户自己的机子处理而已)

- 计划

- 住户授权功能

- 遇到的问题

- Command line is too long: [Idea单元测试报错“Command line is too long” 解决办法](#)  
[AND](#)[IDEA-解决Command line is too long\\_Shorten.....](#)
- test文件红色感叹号, package错误修改: [idea @Test报错, 带有红色感叹号 - 简书\(jianshu.com\)](#)
- Unable to find a @SpringBootConfiguration, you need to use @ContextConfiguration: [SpringBoot测试失败并报错: Unable .....](#)
- Failed to load ApplicationContext: 由这篇文章为灵感[关于Failed to load ApplicationContext报错的原因](#)[AND](#)找到根本上的错误[SpringBoot启动单元测试报错javax.websocket.server.ServerContainer not available](#)
- 多租户分页插件

- 遇到的问题(未解决):

- 查询人员拓展属性
- TenantHandler的getTenantId的where参数什么时候是false(半解决)

TenantHandler 的 getTenantId 方法的 where 参数在某些情况下可能会被设置为 false, 具体取决于实现。

当 where 参数为 false 时, 通常表示不需要对结果进行过滤, 直接返回所有满足条件的记录。例如, 如果 getTenantId 方法的实现是查询数据库中的所有租户ID, 那么 where 参数就可以设置为 false, 不需要对结果进行过滤。

但是, 如果 where 参数为 true, 则表示需要对结果进行过滤, 只返回满足条件的记录。例如, 如果 getTenantId 方法的实现是根据用户ID查询租户ID, 那么 where 参数就需要设置为 true, 以便只返回与指定用户ID相关的租户ID。

因此, where 参数的值取决于具体的实现, 其含义取决于实现中的业务逻辑。

- 笔记

- 循环list: [idea中各种for循环的快捷键](#)
- IDEA自动 import 包: [IDEA开启自动 import 包](#)
- order by 两个字段: [一文讲懂 SQL 排序子句 ORDER BY - 知乎\(zhihu.com\)](#)

## 2023.12.14

- 做了

1. 看接口

- a. estate/projectentrancealarmevent/countByOff: 报警数量接口
- b. estate/projectStaffNotice/countUnRead: 统计未读信息接口  
[SecurityUtils.getAuthentication\(\)](#)涉及到pigcloud的用户村存储
- c. admin/user/getUserDeptRole:
- d. estate/sysCompany/findByGroupOrProjectId/{id}: 通过项目id或项目组id查询集团信息接口
- e. estate/baseBuilding/list: 查询项目下的楼宇列表接口
- f. estate/baseBuilding/listWithGroup: 获取不同项目下的最基本的楼栋列表 (主要是找到哪, 如1栋、2栋、3栋) (写满了我的注释)
- g. [注](#): 上面两个的主要区别是, 搜索出来的房屋名不一样, 并且后者比前者更为简略



- h. estate/projectPersonAttr/{projectId}/{type}/{personId}: 查询对应项目下, 对应类型(如住户、员工、访客)、对应住户id的人员属性接口(但是属性表存储的是什么呢, 阿巴阿巴阿巴阿巴阿巴阿巴)
- i. estate/baseHousePersonRel/pageAll: 分页查询完整的房间住户表(结合住户表和房间住户表)(前面也遇到过,用于住户登记页刚刷新的时候)
- j. estate/baseHousePersonRel/{id}: 根据房间住户表主键id(relaId)查询对应的完整住户信息接口(包括人员基本信息PersonInfo、人员标签信息PersonLabel、重点人员属性FocusPersonAttr、人员属性PersonAttr)
- k. estate/baseBuildingFrame/getCurrentListAndPuid/{entityId}: ①通过指定用户的房间id查出本单元的**其他**房间的列表(房屋框架表, 相同puid就是同单元的房间); ②前端把①查询出来的puid后, 即该用户的单元id, 再传给后端, 所以查出的数据就是和指定用户同栋楼的单元列表(单元可以勉强理解为楼层)
- l. estate/serviceProprietorDevice/{personId}:
- m. estate/projectDeviceInfo/getDeviceByType: 根据设备类型获取对应设备信息列表(在前端表示可通行的设备)(设备(区口机??)信息表)(包括了对应设备类型的设备基本信息, 还有是否可重启, 默认为可以)
- n. estate/projectCard/list/{personId}: 根据住户id查询其所拥有的门禁卡(在门禁卡用户表中搜索)
- o. estate/projectFaceResources/list/{personId}: 通过人员id获取对应的人脸列表

## 2. [自定义typora样式](#)

## 3. [下载配置copyq](#)

### • 计划

- 今天主要看了住户登记页默认接口以及查看住户接口, 所以计划**继续开始看住户授权功能**

### • 遇到的问题

- 加速器要开启了系统代理才能开始测试啊傻逼

### • 遇到的问题(未解决):

- typora设置折叠内容
- estate/serviceProprietorDevice/{personId}: Proprietor业主,所有者
- estate/projectStaffNotice/countUnRead: SecurityUtils.getAuthentication()涉及到pigcloud的用户村存储

### • 笔记:

- Ctrl + Alt+鼠标左键, 点击对应方法, 就能去到实现类
- 自定义typora:
  - [添加自定义 CSS | TyporaChina](#)
  - [自定义列表样式 | TyporaChina](#)
  - [如何DIY你自己的Typora文档](#)
- 下载配置copyq:
  - [CopyQ&Ditto的简单介绍](#)
  - 官网: [CopyQ](#)
  - 下载最新版本(官网下载需要凭证, 所以直接在小网站下载方便): [CopyQ v7.1.0](#)
  - 配置快捷键为: ctrl+alt+`
  - [Win开启剪贴板历史](#)
- typora页内跳转: [Typora页内跳转](#)
- CopyQ基本使用教程: [高效软件10: CopyQ-历史剪贴板管理](#)
- 变量(包括其他相同值)高亮颜色: [idea中设置选中变量\(包括其他相同值\)高亮颜色](#)

- 小结：通行项目主要分为**楼宇管理**（包括楼宇信息表、房屋框架表）、**住户管理**（人员信息表、房间人员表）、**集团管理**（集团表sys\_company、部门（框架）表pigxx.sys\_dept、集团部门表pigxx.sys\_dept\_relation）、**设备管理**（设备（区口机）信息表、门禁卡人员表、人脸人员表）
- 平台->集团->项目组->项目->部门

## 2023.12.15

### • 做了

#### 1. 看接口：

- estate/serviceProprietorDevice/{personId}：通过人员设备表获取人员类型，另外在人员策略表中获取该人员的通行模板/方案(每个人只有一个通行方案)的基本信息(如方案id、生效时间、是否启用等)，这里返回的信息就是该人员id的人员类型、通行方案id、生效时间、是否启用、设备列表（人员设备表中获取）等 **(sql语句好长啊，啊啊啊啊)**
- estate/projectInfo/{id}：通过项目id查询项目组名称（项目组表sys\_project\_group）、集团name（集团表sys\_company）、及项目的电梯是否启用（项目配置表project\_config），**公安是否开启**、以及项目的基本信息（项目基本信息表project\_info）
- estate/servicePassPlan/init：如果通行方案为空(通行方案表pass\_plan)，则初始化人员的通行方案（住户和员工的）基本配置（如：方案名、人员类型、项目id、**业主id???**），并配置不同人员的通行策略
- estate/servicePassPlan/listByType：通过设备类型获取通行方案（传输的基本都是1）
- estate/projectDeviceInfo/getLiftsWithFloor：根据该人员使用的方案id作为设备策略id，还有人员类型、人员id,获取电梯设备列表，内容是**对应用户的的房间所在的楼栋的电梯的楼层数据**（包括了floorVo.setIsDisable("1");// 是否禁止用户改变,1为选中不可变更）
- ★ estate/servicePassPlan/{id}：通过方案id获取相关的完整的方案信息(通行方案名、方案id、方案下的物理设备（通过物理策略id数组获取对应物理设备列表,额外补充 是否可取消选择,默认为不可以）、设备逻辑策略数组、该方案的物理策略的设备ID数组)
- estate/serviceProprietorDevice/listDevice/{personId}/{planId}：通过策略id和personId查询该人员下所有可以使用的设备（区口机、公共区口以及住户所在组团区口机、楼栋梯口机、单元梯口机）

### • 计划

- 细看通行权限=卡片+人脸

### • 遇到的问题

- 可是ProjectPassDeviceVo类都没有disable这个属性[fastjson、Jackson以及Gson序列化对象与get、set以及对象属性之间的关系 fastjson、jackson、gson序列化对象与get、set方法间的关系-CSDN博客](#)

### • 遇到的问题(未解决)

- ~~序列化把get转换为属性问题~~
- estate/projectInfo/{id}：公安是否开启
- estate/servicePassPlan/init，逻辑通行策略宏类型有什么用？？？？

### • 笔记

- copyq开机自启动：win+R打开运行，输入“shell:Common Startup”，将copyQ生成快捷方式放入打开的文件里，即可设置为开机启动
- 是指传进来的实参，和数据表字段无关
- BeanUtils.copyProperties(dto, vo);

- alt+insert: getter和setter构造器快捷键
- 序列化的反射问题: [fastjson、Jackson以及Gson序列化对象与get、set以及对象属性之间的关系](#) [fastjson、jackson、gson序列化对象与get、set方法间的关系-CSDN博客](#)
- [Mybatis常用用法之 sql include的用法 - 掘金 \(juejin.cn\)](#)
- [sql去重复操作详解SQL中distinct的用法-CSDN博客](#)
- [@tableid注解 MyBatis-Plus 常用注解-CSDN博客](#)

## 周小结

我现在大致看了:

- 住户管理 (分页查看住户列表, 新增住户, 查看住户, 修改住户, 授权)
- 集团管理 (集团, 项目组, 项目的管理)
- 房屋管理 (楼栋, 楼层, 房间)
- 消息管理
- 设备管理 (设备, 方案, 策略等)

我后面几天就是打算那个授权的再细看一下, 然后pig4cloud框架也熟悉熟悉, 然后就是还有些接口细节还需要再看看, 大概就是这样 (不包括周末, 哈哈哈哈哈, 周末还要搞课设, 头疼)

## 第三周

2023.12.18

- 做了
  - 为博客添加新域名, 对应新项目 (github无法fork同个项目两次, 所以: 先下载然后修改项目名称->再由webstorm上传项目->修改页面id和项目地址在配置文件中->vercel启动项目并配置id环境->增加自定义项目地址(尽量不要有大写)->腾讯云配置项目地址cname->等凭证弄好就再将该地址改为A类型)
- 计划
- 遇到的问题
- 笔记
  - WebStorm上传代码:
    - 小结: webs中设置git.exe位置、登入github账号+下载git+配置git邮箱账号(与github相同)+webs中设置git share+设置远程文件前缀(分支默认是main,不用管)+点击commit+push
    - [WebStorm中如何将自己的代码上传到github webstorm提交代码到github-CSDN博客](#)
    - [idea中GitHub报错: but initial commit failed: \\*\\*\\* Please tell me who you are. but initial commit failed: author identity unknown-CSDN博客](#)
    - 补充:
      - [稳定可靠的github加速神器 FastGithub - 知乎 \(zhihu.com\)](#)
      - [Git下载安装教程](#)
      - [Git config 查看和设置配置信息](#)
      - [git下载和使用Git上传项目到GitHub](#)

**2023.12.19**

- 做了
- 计划
- 遇到的问题
- 笔记