

分享一点如何在平时的工作中用AI写业务代码！

通常来说我们接到一个需求，可以先梳理成一段话，梳理好后，如果你直接指定对应的代码+你的需求，让ai直接上手改，我发现经常AI是无法完成的，AI只会看你的部分代码，然后你的需求点一次又是好几个，这个时候即使AI正确理解了你的需求，但是写着写着他会忘，目前我在使用Augment、GLM4.5、kimi、cluade来修改我的业务代码哈哈哈，真的是麻了，一个导入的接口，其实逻辑是这样的，在处理excel数据的时候，判断是否有电话、有电话转换成我们内部的供应商+线索数据，然后线索数据只需要去查多一个新表，（我们一个城市下会有多个业务，然后线索这块还得去查这个项目表，然后给每一个城市下面的线索，分别都分每个项目都创建线索）

好的大概逻辑就这样，但是我整体开发下来，人麻了，前前后后不断地换AI来处理，一个不行又换另一个，期间用kimi和glm4.5是因为刚好出来我想体验一下，主要是过程中让他们跑，一开始没掌握技巧，后面看额度烧了不少，没钱了（我是穷逼，只会白嫖哈哈哈）但是那会没有就是对原有已经处理过的逻辑地概念，按道理来说应该就是记录一个文档，类似于记录上一个ai执行到哪里，当前完成了什么业务。类似于一个索引这个文档。不然你换一个AI让AI看代码，ai会大致看，核心逻辑啥的他也分不清楚，类似于一个说明文档，让AI先理解你的需求。所以中途又换来claude，但是呢这边其实也是用的公益中转站，哈哈有的时候用不了人麻了。大概改好了。

由于我们公司就是这个接口本地调不了，然后我跑那个单元测试，mock数据也感觉挺麻烦，每次就直接推开发环境测试，

今天用了一下augment，感觉这个没有这么智能了，最近用起来感觉就是一个智障，效果一点也不好，可能我太贪快了，你问他一个问题，他就解决了当前这个问题，然后其他的问题又没有解决连续的，基本上提什么就解决啥问题，就是解决问题的思路不对，思路不连贯。这种情况我发现其实是挺致命的。如果有一个代码有连续的逻辑，比方说有四个连贯逻辑，你会发现ai指只会改一开始的逻辑，后面逻辑就不连贯了，然后你在提问，ai再改，这样下去，肯定你就不想问了，你觉得他是傻逼，你越会觉得ai不能写业务代码。

1、当前的---品类映射（海外数据使用默认品类）这块不对！！

中国：保持“一级匹配 + 取第一个二级”的历史行为

非中国：使用 secondary\_category（一级品类中文名）+ secondary\_category\_lv2（二级品类中文名）精确匹配对应 ID；二级不命中则置空（不回退），具体的处理逻辑通过 som\_commercial\_category 表去查库，匹配拿到 id 应该是 - 和中国处理逻辑类似，只不过这块一级二级品类取得是传过来的

2、确认一下非中国：normalizeOverseasTel（仅 trim，不校验），一般数据只会有一个电话，反正就是直接把电话插入进去即可，当前是否市这么实现？

3、/\*\*

\* 二级品类

\*/

@ExcelProperty(value = "secondary\_category\_lv2", index = 9)

private String secondaryCategoryLv2;

这块参数我已经补充到 AiBrandImport 这里了

注意！不要改变原有的导入逻辑！！

补充大概的表名：

```
CREATE TABLE `som_commercial_category` (  
  `id` bigint NOT NULL COMMENT '主键id',  
  `parent_id` bigint NOT NULL DEFAULT '0' COMMENT '父级Id',  
  `parent_fname` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci DEFAULT NULL  
  COMMENT '全路径名称',  
  `parent_fid` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci DEFAULT NULL  
  COMMENT '父级完整ID',  
  `category_name` varchar(32) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci DEFAULT NULL  
  COMMENT '类别名,最多输入50字',  
  `slevel` int DEFAULT '0' COMMENT '层级,根节点为0',  
  `sort_num` int DEFAULT '0' COMMENT '排序',  
  `create_by` bigint DEFAULT NULL COMMENT '创建人id',  
  `create_user` varchar(32) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci DEFAULT NULL  
  COMMENT '创建人账号/手机号',  
  `create_user_name` varchar(32) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci DEFAULT NULL  
  COMMENT '创建人姓名/昵称',
```

// 4. 二级品类未找到，返（不回退到一级）

CommercialCategoryListDto secondaryDto2 = new CommercialCategoryListDto();

secondaryDto.setParentId(primaryId);

secondaryDto.setSlevel(2);

List<CommercialCategoryListVo> secondaryList2 = commercialCategoryProvider.list(secondaryDto2);

return secondaryList2.get(0).getId();

} catch (Exception e) {

log.error("查询海外品类失败 - 一级: '{}', 二级: '{}'", primaryCategory, secondaryCategory, e);

return null;

}

这块我稍微修改一下，二级品类未找到，返回该一级品类下的第一个二级品类，就和中国处理逻辑一样，这样行吗？还有哪里需要修改一下

1、我发现要一个问题，就是当前方法中存在很多没啥用的方法，就是没有被调用的，请把这些方法删了，然后确保主逻辑不变！！  
2、当前这个：String cleanTel = isChinaCountry(dto.getCountry()) ? processPhoneNumber(dto.getTel()) : normalizeOverseasTel(dto.getTel());方法我发现我即使传了越南的国家的，走的还是中国判断的逻辑，所以先执行上面的逻辑，然后把刚刚修改的逻辑加上，目前电话这块逻辑没走到！！

日志的主逻辑还是走的原有的中国逻辑

```
2025-08-18 10:40:58.532 [import-1] INFO c.s.s.c.s.i.AiBrandImportServiceImpl -  
[processPhoneNumber,1037] - [电话号码处理] 开始处理电话号码，原始  
值: '84966396698'  
2025-08-18 10:40:58.532 [import-1] WARN c.s.s.c.s.i.AiBrandImportServiceImpl -  
[processPhoneNumber,1087] - [电话号码处理] ⚠ 号码片段1格式无效 - 原  
始: '84966396698' → 清洗后: '84966396698'  
2025-08-18 10:40:58.532 [import-1] WARN c.s.s.c.s.i.AiBrandImportServiceImpl -  
[processPhoneNumber,1109] - [电话号码处理] ✗ 处理失败 - 所有号码片段都无效，原  
始: '84966396698' → 返回null  
2025-08-18 10:40:58.551 [import-1] INFO c.s.s.c.s.i.AiBrandImportServiceImpl -  
[processPhoneNumber,1037] - [电话号码处理] 开始处理电话号码，原始  
值: '84392257292'  
2025-08-18 10:40:58.552 [import-1] WARN c.s.s.c.s.i.AiBrandImportServiceImpl -  
[processPhoneNumber,1087] - [电话号码处理] ⚠ 号码片段1格式无效 - 原  
始: '84392257292' → 清洗后: '84392257292'  
2025-08-18 10:40:58.552 [import-1] WARN c.s.s.c.s.i.AiBrandImportServiceImpl -  
[processPhoneNumber,1109] - [电话号码处理] ✗ 处理失败 - 所有号码片段都无效，原  
始: '84392257292' → 返回null
```

10:58 AM

1、我发现要一个问题，就是当前方法中存在很多没啥用的方法，就是没有被调用的，请把这些方法删了，然后确保主逻辑不变！！  
2、当前这个：String cleanTel = isChinaCountry(dto.getCountry()) ? processPhoneNumber(dto.getTel()) : normalizeOverseasTel(dto.getTel());方法我发现我即使传了越南的国家的，走的还是中国判断的逻辑，所以先执行上面的逻辑，然后把刚刚修改的逻辑加上，目前电话这块逻辑没走到！！

日志的主逻辑还是走的原有的中国逻辑

```
2025-08-18 10:40:58.532 [import-1] INFO c.s.s.c.s.i.AiBrandImportServiceImpl -  
[processPhoneNumber,1037] - [电话号码处理] 开始处理电话号码，原始  
值: '84966396698'  
2025-08-18 10:40:58.532 [import-1] WARN c.s.s.c.s.i.AiBrandImportServiceImpl -  
[processPhoneNumber,1087] - [电话号码处理] ⚠ 号码片段1格式无效 - 原  
始: '84966396698' → 清洗后: '84966396698'  
2025-08-18 10:40:58.532 [import-1] WARN c.s.s.c.s.i.AiBrandImportServiceImpl -  
[processPhoneNumber,1109] - [电话号码处理] ✗ 处理失败 - 所有号码片段都无效，原
```

```

// 检查是否是从基础验证传来的海外数据标记
if ("OVERSEAS_PENDING".equals(dto.getMappedCategory())) {
    // 海外数据：使用 secondary_category（一级）+ secondary_category_lv2（二级）精确匹配
    Long categoryId = findOverseasCategoryId(dto.getSecondaryCategory(),
dto.getSecondaryCategoryLv2());
    if (categoryId == null) {
        log.error("[数据验证] X 海外品类映射失败 - 记录: {}, 一级品类: '{}', 二级品类: '{}'",
            recordId, dto.getSecondaryCategory(), dto.getSecondaryCategoryLv2());
        result.addFailure(standardizedBrandName, "未找到海外品类映射: " +
dto.getSecondaryCategory() +
            (StringUtils.hasText(dto.getSecondaryCategoryLv2()) ? " -> " + dto.getSecondaryCategoryLv2()
: ""));
        return false;
    }
    dto.setMappedCategory(categoryId.toString());
    log.info("[数据验证] ✓ 海外品类映射成功 - 记录: {}, 一级品类: '{}', 二级品类: '{} → 品类ID: {}",
        recordId, dto.getSecondaryCategory(), dto.getSecondaryCategoryLv2(), categoryId);
} else if (isChinaCountry(dto.getCountry())) {
    // 中国：保持"一级匹配 + 取第一个二级"的历史行为
    String mappedCategory = CATEGORY_MAPPING.get(dto.getSecondaryCategory());
    if (mappedCategory == null) {
        log.error("[数据验证] X 品类映射失败 - 记录: {}, 外部品类: '{}', 无对应的内部品类",
            recordId, dto.getSecondaryCategory());
        result.addFailure(standardizedBrandName, "未找到品类映射: " + dto.getSecondaryCategory());
        return false;
    }
    dto.setMappedCategory(mappedCategory);
    log.info("[数据验证] ✓ 品类映射成功 - 记录: {}, 外部品类: '{} → 内部品类: '{}',
        recordId, dto.getSecondaryCategory(), mappedCategory);
} else {
    // 非中国：使用 secondary_category（一级）+ secondary_category_lv2（二级）精确匹配，未找到
    则取一级下第一个二级
    Long categoryId = findOverseasCategoryId(dto.getSecondaryCategory(),
dto.getSecondaryCategoryLv2());
    if (categoryId == null) {
        log.error("[数据验证] X 海外品类映射失败 - 记录: {}, 一级品类: '{}', 二级品类: '{}'",
            recordId, dto.getSecondaryCategory(), dto.getSecondaryCategoryLv2());
        result.addFailure(standardizedBrandName, "未找到海外品类映射: " +

```

11:30 AM

不对最后一条数据不是越南的，所以当前这个方法判断不对-你的逻辑没有部分工作，就是存在问题  
这块不对啊即使这样改也只有一个一级品类，但是对应的二级品类也没有实现，现在一级二级品类id和名称都要塞进去，在越南的数据

您说得对！我重新分析一下问题：

插入到som\_commerce\_brand这个表中啊，这个方法不是已经有了对应的值了吗设置进去就好了

```
/**
 * 查询海外品类ID
 * @param primaryCategory 一级品类名称
 * @param secondaryCategory 二级品类名称
 * @return 品类ID，如果未找到返回null
 */
private Long findOverseasCategoryId(String primaryCategory, String secondaryCategory) {
    try {
        // 1. 先查找一级品类
        CommercialCategoryListDto primaryDto = new CommercialCategoryListDto();
        primaryDto.setCategoryName(primaryCategory);
        primaryDto.setSlevel(1);
        List<CommercialCategoryListVo> primaryList = commercialCategoryProvider.list(primaryDto);

        if (CollectionUtils.isEmpty(primaryList)) {
            return null;
        }

        Long primaryId = primaryList.get(0).getId();

        // 2. 如果没有二级品类，直接返回一级品类ID
        if (!StringUtils.hasText(secondaryCategory)) {
            return primaryId;
        }
    }
}
```

```
// 检查是否是从基础验证传来的海外数据标记
if ("OVERSEAS_PENDING".equals(dto.getMappedCategory())) {
    // 海外数据：使用 secondary_category（一级）+ secondary_category_lv2（二级）精确匹配
    OverseasCategoryInfo categoryInfo = findOverseasCategoryInfo(dto.getSecondaryCategory(),
        dto.getSecondaryCategoryLv2());
    if (categoryInfo == null) {
        log.error("[数据验证] X 海外品类映射失败 - 记录: {}, 一级品类: '{}', 二级品类: '{}',
            recordId, dto.getSecondaryCategory(), dto.getSecondaryCategoryLv2());
        result.addFailure(standardizedBrandName, "未找到海外品类映射: " +
            dto.getSecondaryCategory() +
            (StringUtils.hasText(dto.getSecondaryCategoryLv2()) ? " -> " + dto.getSecondaryCategoryLv2()
                : ""));
        return false;
    }
    // 将品类信息编码到mappedCategory字段中，格式：
    primaryId|primaryName|secondaryId|secondaryName
    String encodedCategory = String.format("%d|%s|%s|%s",
        categoryInfo.getPrimaryId(),
        categoryInfo.getPrimaryName(),
        categoryInfo.getSecondaryId() != null ? categoryInfo.getSecondaryId().toString() : "",
        categoryInfo.getSecondaryName() != null ? categoryInfo.getSecondaryName() : "");
    dto.setMappedCategory(encodedCategory);
    log.info("[数据验证] ✓ 海外品类映射成功 - 记录: {}, 一级品类: '{} (ID:{})', 二级品类: '{} (ID:{})',
        recordId, categoryInfo.getPrimaryName(), categoryInfo.getPrimaryId(),
        categoryInfo.getSecondaryName(), categoryInfo.getSecondaryId());
}

// 5. 验证和映射品类（支持海外数据）
if (isChina) {
    // 中国：使用映射表
    String mappedCategory = CATEGORY_MAPPING.get(dto.getSecondaryCategory());
}
```



```
// 如果是编码的海外品类信息，验证格式
if (dto.getMappedCategory().contains("|")) {
    String[] parts = dto.getMappedCategory().split("\\|");
    if (parts.length < 2) {
        log.error("[数据验证] X 海外品类信息格式错误 - 记录: {}, 品类信息: '{}'", recordId,
dto.getMappedCategory());
        result.addFailure(standardizedBrandName, "海外品类信息格式错误");
        return false;
    }
    log.info("[数据验证] ✓ 海外品类映射验证通过 - 记录: {}, 一级品类: '{}', 二级品类: '{}'",
        recordId, parts[1], parts.length > 3 ? parts[3] : "无");
} else {
    log.info("[数据验证] ✓ 中国品类映射验证通过 - 记录: {}, 品类: '{}'", recordId,
dto.getMappedCategory());
}
}
}

}这块你确定对于原有的逻辑不会改吗？对中国处理逻辑没影响吗
原又代码不是这样的吗： // 中国：保持"一级匹配 + 取第一个二级"的历史行为
String mappedCategory = CATEGORY_MAPPING.get(dto.getSecondaryCategory());
if (mappedCategory == null) {
    log.error("[数据验证] X 品类映射失败 - 记录: {}, 外部品类: '{}', 无对应的内部品类",
        recordId, dto.getSecondaryCategory());
    result.addFailure(standardizedBrandName, "未找到品类映射: " + dto.getSecondaryCategory());
    return false;
}
dto.setMappedCategory(mappedCategory);
log.info("[数据验证] ✓ 品类映射成功 - 记录: {}, 外部品类: '{}' → 内部品类: '{}',
    recordId, dto.getSecondaryCategory(), mappedCategory);
你再评估一下
```

最后我用ai整理一下：当前全局得问题，然后再交给ai

任务标题：修复：海外数据在创建supplier\_brand关联时因品类解析错误导致数据库异常

模块/文件：AiBrandImportServiceImpl.java

问题描述：

在导入海外品牌数据时，程序在createSupplierBrandRelation方法内创建供应商与品牌关联记录（som\_commerce\_supplier\_brand）时发生数据库写入失败。日志显示Data truncation: Data too long for column 'category'错误。

根本原因分析：

数据格式不一致：对于海外数据，dto.getMappedCategory()字段中存储的是一个经编码的字符串，格式为“主ID|主名称|次ID|次名称”（例如：“123|越南菜|456|河粉”）。

解析逻辑缺失：createSupplierBrandRelation方法在处理此字段时，未像buildBrandAddDto方法一样对其进行解析。它错误地将整个编码字符串直接赋给了category字段，导致字符串超长。

关联错误：由于未解析，后续调用getCategoryMappingInfo()也无法找到匹配项，导致category\_id字段最终为null。

预期行为：

成功导入海外数据后，som\_commerce\_supplier\_brand表中对应的记录应为：

category字段：存储一级品类名称（例如：“越南菜”）