后端Java项目AI编程规则

springboot单项目，对应人人框架企业版单体服务版

```
# 项目规则说明

## 项目技术栈
- 基础技术栈: Java 17、Spring Boot、mybatis、MyBatis-Plus
- 数据库: MySQL 5.7
- 缓存: Redis 5.0
- 消息队列: RabbitMQ 3.8
- 日志: SLF4J、Logback
- 工具类: Lombok、Hutool
- 构建工具: Maven

## 文件目录结构
- `sketch-viewer-admin`: spring boot 项目，为管理员界面提供RESTful API接口，包含Controller、DTO、VO、Mapper、Service、ServiceImpl、exception、interceptor
- `sketch-viewer-api`: spring boot 项目，为APP界面提供RESTful API接口，包含Controller、DTO、VO、Mapper、Service、ServiceImpl、exception、interceptor
- `renren-common`: spring boot 项目，公共模块，禁止直接修改本文件夹的任何文件，包括子文件夹
- `renren-dynamic-datasource`: spring boot 项目，动态数据源模块，禁止直接修改本文件夹的任何文件，包括子文件夹
- `doc`: 文件夹，禁止直接修改本文件夹的任何文件，包括子文件夹
- `docker`: 文件夹，禁止直接修改本文件夹的任何文件，包括子文件夹


## sketch-viewer-admin 目录
- spring boot 项目，使用maven构建
- 本项目虽然支持多种数据库，但默认使用MySQL 5.7数据库，暂时不需要支持其他数据库。
- db 目录: 禁止直接修改本文件夹的任何文件，包括子文件夹
- db2 目录: 在数据库设计阶段，所有生成的业务数据库表结构定义文件必须统一存储在项目根目录下的 `db2` 目录中，针对每个业务模块，需在 `db2` 目录下创建对应的子目录，子目录名称与业务模块名称保持一致（采用小
- src/main/java/io/renren/modules/demo: 这是一个示例模块，新添加的模块都应该严格参考此模块的结构、命名规范、接口定义、实现示例。
- java开发规范
    - 类命名规范: 采用大驼峰命名法，如 `OrderManagementController`。
    - 方法命名规范: 采用小驼峰命名法，如 `getOrderDetail`。
```

- 变量命名规范：采用小写蛇形命名法，如 `order_id`。
- 常量命名规范：采用全部大写字母，单词之间用下划线隔开，如 `MAX_ORDER_COUNT`。
- Controller 层的 RESTful API 接口规范
    - 禁止在Controller层直接操作数据库。所有与数据库相关的操作必须通过Service层完成。Controller层的职责仅限于处理HTTP请求、调用Service层的方法，并将结果返回
Object）。
    - Controller 层的 RESTful API
接口必须使用POST方法（@PostMapping），参数必须通过请求体（@RequestBody）传递，禁止使用GET方法传递业务参数。禁止使用PUT（@PutMapping）、DELETE（@Dele
    - Controller 层的 RESTful API 接口必须返回JSON格式的响应体，必须使用io.renren.common.utils.Result<T>
类封装响应体，其中T为接口调用结果数据的类型。io.renren.common.utils.Result<T> 类的定义如下：

        ```java
        public class Result<T> {
            private int code;
            private String msg;
            private T data;
        }
        ```

- 数据操作规范
    - 所有数据操作都必须使用MyBatis-Plus提供的CRUD方法，仅在必要时才使用MyBatis的Mapper接口。
    - 数据库操作实现应该优先考虑使用MyBatis-Plus提供的CRUD方法，仅在必要时才使用MyBatis的Mapper接口。

- 数据库设计规范
    - 数据表命名规范：采用小写蛇形命名法，如 `order_management`。
    - 数据表字段命名规范：采用小写蛇形命名法，如 `order_id`。
    - 数据表都必须包含一个主键字段，主键字段命名为 `id`，数据类型为 `BIGINT`，并设置为自增。
    - 数据表必须包含一个 `create_date` 字段，数据类型为 `datetime`，用于记录数据创建时间。
    - 数据表必须包含一个 `update_date` 字段，数据类型为 `datetime`，用于记录数据最后更新时间。
    - 数据表必须包含一个 `creator` 字段，数据类型为 `bigint`，用于记录数据创建人ID。
    - 数据表必须包含一个 `updater` 字段，数据类型为 `bigint`，用于记录数据最后更新人ID。
    - 数据表必须包含一个 `delete_flag` 字段，数据类型为 `tinyint(1)`，默认值为 `0`，用于记录数据是否被逻辑删除，0表示未删除，1表示已删除。

```

版本号 #7
由 董列涛 创建于 4 十二月 2025 10:59:11
由 董列涛 更新于 9 一月 2026 11:42:20
```