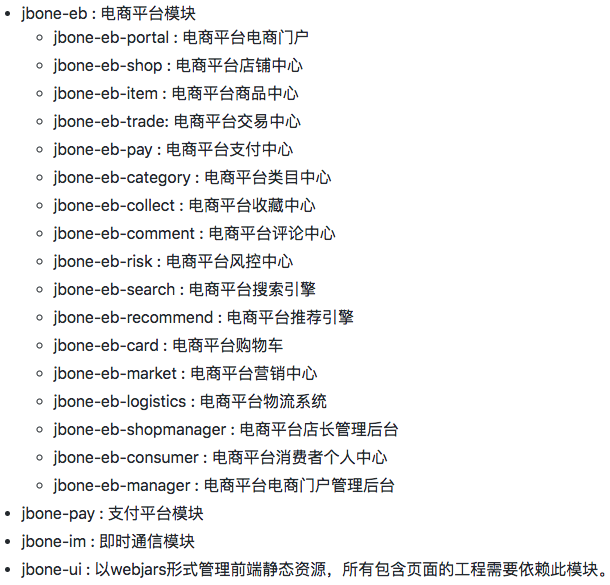
**Ace项目开发部署规范**

# 现有项目介绍





# 项目结构规范

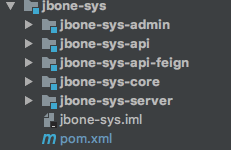
## 一级项目标准

满足以下条件才能设为一级项目：

1. 完全独立于其他模块的功能
2. 其他模块共同依赖，如common、configuration、tag
3. 功能类型相同的项目，可以组合起来设立一个一级项目，如cas、sm

## 项目结构规范

以系统管理模块为例：



core：系统核心，包含业务逻辑层和数据访问层。

server：对外提供的微服务，可独立打包运行，注册到注册中心供其他服务调用。

api: 对外接口定义，本模块中只提供接口定义和model。

api-starter：对外接口的Feign配置，调用端只需要添加此依赖，就完成了配置，不需要其他额外的代码实现。

admin: 管理后台，一般情况下每个模块都要对应一个管理后台，供管理员维护和管理系统。

front：前台展示模块，主要面向外部用户。

**依赖关系**

admin -> core

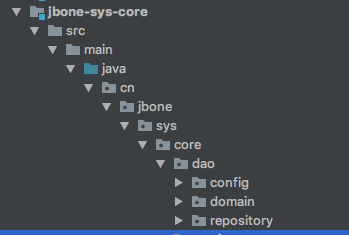
server-> core

->api

**特别注意，项目之间的依赖除了客户端等公共Jar（如jbone-cas-client）以外，只允许依赖api和api-feign模块，不能有其他的依赖。很好理解，service等其他模块是微服务的自己的，不应该耦合在一块的。Api模块是微服务的客户端封装，所以要使用其他服务的功能的话，需要引入API，然后调用。**

## 代码结构规范

**核心模块DAO包规范**

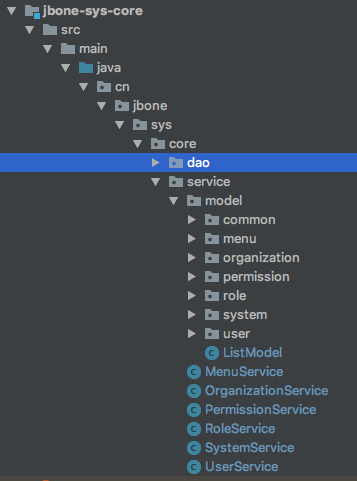


config：数据源配置

domain：ORM实体类

repository：JPA接口定义

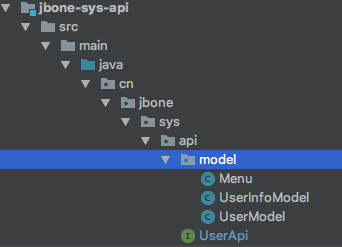
**核心模块Service包规范**



model：对外返回的model定义（对外返回的信息不等于ORM实体类定义，所以这里要搞一层返回结果model）

xxxService：具体业务处理类

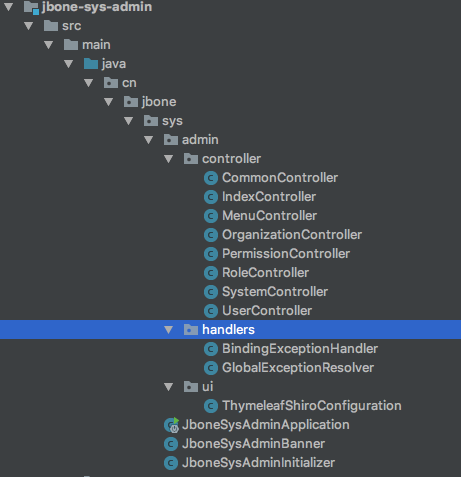
**API模块规范**



model：调用端引用的model（应用在调用端引用）

api：接口定义

**admin模块规范**



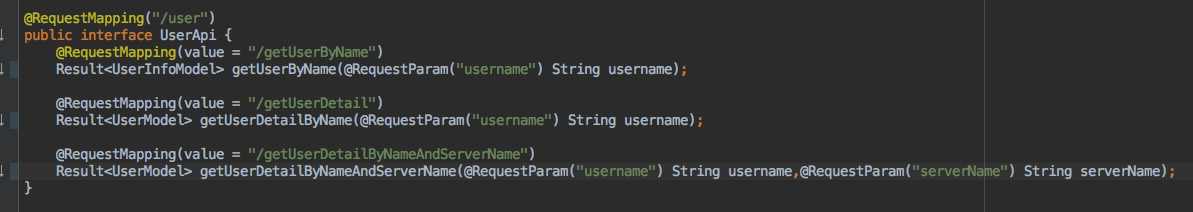
controller：对应mvc的C实现

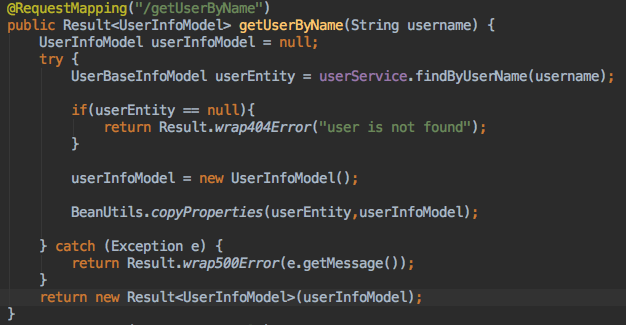
handler：一些异常和其他的处理器

## 远程调用规范

Aceberg项目中的远程调用默认都是用starter实现。

接口定义按以下规范：





使用Result<T>作为返回结果，T为API中定义的model。实现时一定要注意异常的返回处理，不要把内部的异常再抛给调用端了。

调用端调用时使用result.isSuccess（）来判断服务是否正常返回。

## 实体类定义规范

VO – 页面展示Bean

DTO – RPC传输Bean

XXXRequestDTO – RPC远程入参

XXXResponseDTO – RPC远程出参

Entity – ORM映射Bean



各Bean在系统中的传输过程

## 代码命名规范和注释

命名

1. 见名知意
2. 驼峰命名

代码注释

1. 类上必须加注释，说明作用和功能
2. 方法必须加注释，说明功能
3. 核心逻辑必须加注释，说明业务逻辑
4. 全局变量、常量、配置属性必须加注释