



HZERO 培训

中台技术中心 - 伯姜洲

上海· 2019年07月

基础开发规范

◆ 版本规范

- 使用语义化版本控制
- 版本号递增规则如下
 - 主版本号：当你做了不兼容的 API 修改，递增主版本号
 - 次版本号：当你做了向下兼容的功能性修改，递增次版本号
 - 修订号：当你做了向下兼容的问题修正，递增修订号

◆ 命名规范

- 值集数据配置
- LOV应用配置
- 编码规则配置
- 消息模板代码
- 多语言描述维护
- 返回消息

◆ 链接

- 基础开发规范：<http://hzerodoc.saas.hand-china.com/zh/docs/development-specification/basic/>

工程结构

◆ 命名规范

- 项目从大到小可分为产品、服务、组件，产品以产品名称命名，如hzero、oa、srm。
- 一个产品下划分多个服务，命名规则：产品名-模块名，如hzero-platform、oa-contract。
- 如果产品下有组件，命名格式：产品名-boot-组件名，如hzero-boot-platform、hzero-boot-common。

◆ GVA的定义

- GroupID 格式： org.[组织/BU].[子业务线]，最多 4 级。
 - 正例： org.hzero 或 org.hzero.boot
- ArtifactID 格式： 产品线名-模块名。语义不重复不遗漏，先到中央仓库去查证一下。
 - 正例： hzero-platform / hzero-mdm / hzero-boot-common
- Version： 初始化版本定义为0.1.0-SNAPSHOT

◆ 应用分层

- HZERO应用分层采用 DDD 代码结构，请参考文档。

◆ 链接

- 工程结构： <http://hzerodoc.saas.hand-china.com/zh/docs/development-specification/backent-development-specification/structure/>

Java基础规范-代码规范

◆ 规范检查&工具设置

- IDE 开启阿里巴巴规范检查
- IDE 安装代码格式化插件
- IDE 工具设置

◆ 代码规约

- 熟悉 《[阿里巴巴Java开发手册](#)》

◆ 注释规约

- 类注释
- 方法注释
- 字段注释

◆ 链接

- Java基础规范: <http://hzerodoc.saas.hand-china.com/zh/docs/development-specification/backent-development-specification/basic/>

Java基础规范-异常处理

◆ 异常类型

- debug: 非常具体的信息, 用于开发调试使用
- info: 重要的业务处理记录信息
- warn: 发生这个级别问题时, 处理过程可以继续, 但必须对这个问题给予额外关注
- error: 系统中发生了非常严重的问题, 必须马上有人进行处理

◆ 异常规范

- `CommonException`: 抛出程序业务异常
- `FeignException`: 内部接口调用异常
- `Logger` 打印日志, 使用占位符的形式, 不要字符串拼接
- 不允许出现 `System.out.println`
- 不允许出现 `e.printStackTrace()`
- 不允许记录日志后又抛出异常, 因为这样会多次记录日志, 只允许记录一次日志, 应尽量抛出异常, 顶层打印一次日志。

Java基础规范-RestApi-1

◆ 设计原则

- URI末尾不要添加 "/"
- 禁止在URL中使用 "_"
- 禁止使用大写字母
- 不要在URI中包含扩展名
- 建议URI中的名称使用复数
- 建议URI设计时只包含名词，不包含动词
- 租户级URI设计时需要包含租户ID

◆ HTTP动词

- GET (SELECT) : 从服务器取出资源（一项或多项）。
- POST (CREATE) : 在服务器新建一个资源。
- PUT (UPDATE) : 在服务器更新资源（客户端提供改变后的完整资源）。
- PATCH (UPDATE) : 在服务器更新资源（客户端提供改变的属性）。
- DELETE (DELETE) : 从服务器删除资源。

◆ HTTP响应设计

- 2xx: 成功类别
- 3xx: 重定向类别
- 4xx: 客户端错误类别
- 5xx: 服务器错误类别

Java基础规范-RestApi-2

◆ API版本管理

- 建议通过URI指定服务版本，版本采用字符“v”+数字主版本号，例如，/v1/xxxs
- 服务后端分包：controller.v1/controller.v2
- API升级建议：尽量做兼容性设计，不要随便升级版本

◆ Action 命名规范

- list: 查询所有
- query: 获取单个资源
- create: 创建单个资源
- update: 更新单个资源
- delete: 删除单个资源

◆ 数据响应

- 数据统一使用 ResponseEntity 封装返回到前端
- 前端通过判断 failed=true 来判断是否失败，请求成功则没有 failed 属性
- 后端必须对传入的数据做合法性校验，可以包装异常抛出，程序会自动捕获
- 程序自主抛出的异常统一使用 CommonException 封装对应的错误码抛出
- 程序中可预见的及捕获的异常尽量封装成 CommonException 抛出，由程序自动处理
- 前端应该对数据做好校验工作，按正常情况，视前端到后端的数据都是合法的

数据库规约-1

◆ 命名规范

- 库名使用服务名命名，比如hzero_shceduler、hzero_platform
- 服务级表规范：服务简码_表名，服务简码4位，表名最多4段，总共不超过5段。如IAM服务的module表：hiam_module
- 多语言表：数据多语言表，在基表的基础上，以_tl结尾。如LOV值多语言表hpfm_lov_value_tl

◆ 表设计规范

- 表名长度不要超过26个字符，否则Oracle建相关对象可能会报错
- 列名称长度不允许超过30个字符，否则Oracle建相关对象可能会报错

◆ 索引规范

- 主键索引名为 表名_pk；唯一索引名为 表名_u1..5；普通索引名则为 表名_n1..5。
- 单张表索引数量建议控制在5个以内
- 组合索引字段数不建议超过5个

◆ 链接

- 数据库规约：<http://hzerodoc.saas.hand-china.com/zh/docs/development-specification/backent-development-specification/database/>

数据库规约-2

◆ 表更新规范

- 为了便于服务升级、发布时表结构自动更新，必须规范表结构更新规范。总体要求不能没有任何记录而直接在数据库执行DDL语句更改表结构，必须同步更新 Excel 表设计，然后添加 liquibase groovy 脚本 changeSet 来更新表结构，这样 DATABASECHANGELOG 表中才能保留更新记录。
- 第一步，无论是新增表、还是增删改字段，先更新 Excel 表设计。
- 第二步，更新数据库 liquibase groovy 脚本
- 第三步，使用初始化工具更新开发库表结构
- 第四步，升级时，使用初始化工具更新其他环境表结构

Swagger描述规范

◆ Swagger 规范

- 编写良好的 Swagger 描述信息有助于前后端API联调测试
- @Api(tags = SwaggerApiConfig.TASK): 标注在 Controller 类上, 描述类信息
- @ApiOperation(value = "xxx"): 标注在 Controller 接口方法上, 描述API层级、描述信息、访问控制等
- @ApiModel("xxx"): 标注在实体类上, 描述实体信息
- @ApiModelProperty("xxx"): 标注在实体字段上, 描述字段信息
- @ApiImplicitParams(): 标注在 Controller 接口方法上, 对接口参数进行说明
- @ApiParam(value = "xxx"): 标注在 Controller 接口方法参数上, 对参数进行说明

◆ 链接

- Swagger 描述规范: <http://hzerodoc.saas.hand-china.com/zh/docs/development-specification/backent-development-specification/swagger/>

单元测试

◆ Spock

- 内置支持 mocking stubbing, 可以很容易地模拟复杂的类的行为
- 兼容性强, 内置 Junit Runner, 可以像运行 Junit 那样运行 Spock, 甚至可以在同一个项目里面同时使用两种测试框架
- Spock 代码风格简短, 易读, 表达性强, 扩展性强, 还有更清晰显示bug

◆ 链接

- 单元测试: <http://hzerodoc.saas.hand-china.com/zh/docs/development-specification/backent-development-specification/unit-test/>

代码生成器

◆ 代码生成器

- 根据表生成基本的DDD模型的代码
- 根据表生成 groovy 脚本

◆ 链接

- 代码生成器: <http://hzerodoc.saas.hand-china.com/zh/docs/development-guide/generator/backend/>

课后作业

- ✓ 在 HZERO 文档平台上熟悉开发规范，重点关注PPT中提到的相关规范。