公司或组织名称：逐一科技

时间：2018年11月06日

**字典管理模块企划**

# 数据库

## 表结构

CREATE TABLE sys\_dictionary\_type(

id bigint(20) NOT NULL AUTO\_INCREMENT,

parentId bigint(20) NOT NULL COMMENT'上级ID'

name varchar(255) NOT NULL COMMENT'名称',

genre tinyint(1) NOT NULL COMMENT'类别：0.系统结构,1.字典类别'

order tinyint(4) DEFAULT NULL COMMENT '排序',

creatorUserId bigint(20) DEFAULT NULL COMMENT '创建者',

creationTime datetime DEFAULT NULL COMMENT '创建时间',

lastModifierUserId bigint(20) DEFAULT NULL COMMENT '更新者',

lastModificationTime datetime DEFAULT NULL COMMENT '更新时间',

deleterUserId bigint(20) DEFAULT NULL COMMENT '删除者',

deletionTime datetime DEFAULT NULL COMMENT '删除时间',

deleted tinyint(1) DEFAULT '0' COMMENT '删除标记',

version bigint(20) DEFAULT '1' COMMENT '版本号',

PRIMARY KEY (id),

KEY inx\_name (name)

)ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8 COMMENT='字典类型管理表';

CREATE TABLE sys\_dictionary (

id bigint(20) NOT NULL AUTO\_INCREMENT,

typeId bigint(20) NOT NULL COMMENT'类型ID'

tenantId bigint(20) DEFAULT NULL COMMENT'租户ID',

name varchar(255) NOT NULL COMMENT'项目名',

value varchar(255) NOT NULL COMMENT '项目值',

logogram varchar(255) NOT NULL COMMENT '简拼',

order tinyint(4) NOT NULL COMMENT '排序',

status tinyint(1) DEFAULT '0' COMMENT '是否有效：0.无效 1.有效',

remarks varchar(255) DEFAULT NULL COMMENT '备注',

creatorUserId bigint(20) DEFAULT NULL COMMENT '创建者',

creationTime datetime DEFAULT NULL COMMENT '创建时间',

lastModifierUserId bigint(20) DEFAULT NULL COMMENT '更新者',

lastModificationTime datetime DEFAULT NULL COMMENT '更新时间',

deleterUserId bigint(20) DEFAULT NULL COMMENT '删除者',

deletionTime datetime DEFAULT NULL COMMENT '删除时间',

deleted tinyint(1) DEFAULT '0' COMMENT '删除标记',

version bigint(20) DEFAULT '1' COMMENT '版本号',

PRIMARY KEY (id),

KEY idx\_value (value),

KEY inx\_name (name),

KEY idx\_tenantId (tenantId)

) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8 COMMENT='字典条目管理表';

## 建表思路

一共两张表,一张是字典类别表(上级)

一张则是字典条目表。(商户自定义)

# 二、接口计划

## 1.接口规划-规划依据

(1.根据typeId来确定条目的字典类型归属

(2.字典管理分两个片段

左边根据系统结构来进行字典分类

右侧则为分类下的字典条目

(3.字典类别数据由后台代码生成和迭代

字典条目表利用typeId与子弟那类别表关联

基础条目由枚举表示

## 2.接口方案-编号递增

1.addDictionaryEntry:

添加字典条目

2.deleteDictionary:

删除字典条目

3.updateDictionary:

修改字典条目

4.getDictionaryStatus：

获取字典条目的状态是否有效：0.无效 1.有效

5.getDictionaryChildPagedList：

根据点击获取的typeId来分页获取下属条目

6.getDictionaryPagedList：

根据查询条件分页获取字典条目

7.dictionaryTypeTree:

获取字典类别树

## 计划周期

代码编写:2天

代码测试:1天

代码调试:1天

总计时间:4天

## 4.业务疑问

基础数据枚举的处理