# 概述

## 业务分析

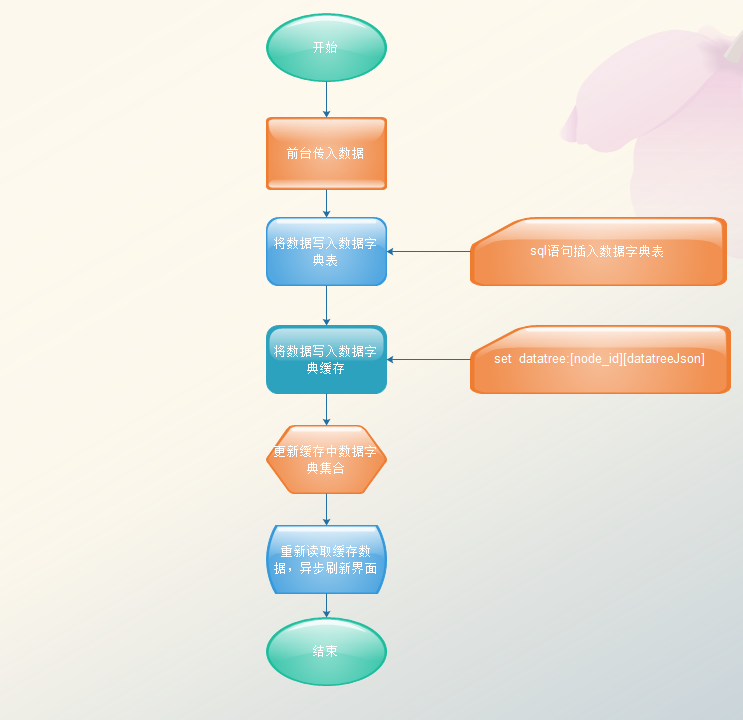
ECMC增、删、改、同步操作，同时需要维护MySQL数据表和Redis缓存，而查询操作，依然查询数据库。而ECSC仅需要提供接口操作Redis缓存即可

## ECSC流程图

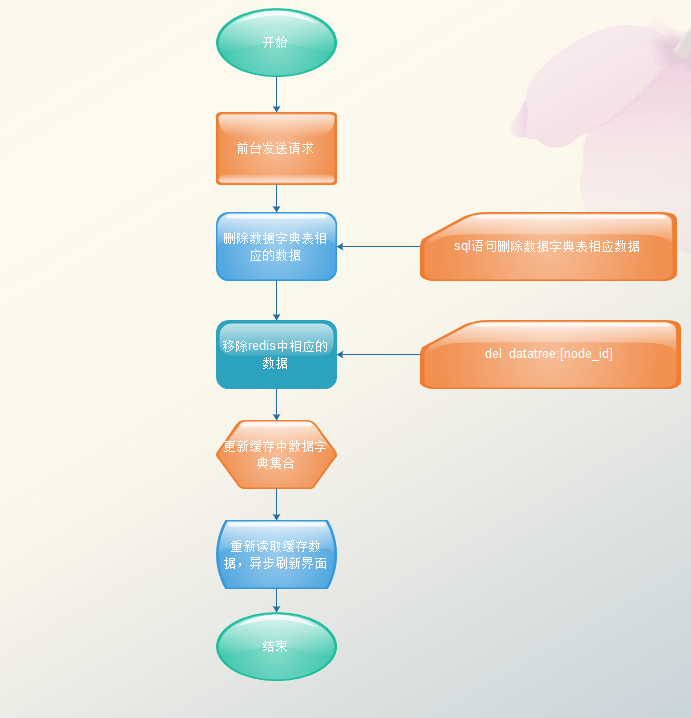
ECMC通过同步功能将数据初始化到Redis中，ECSC直接通过接口使用。

## ECMC流程图

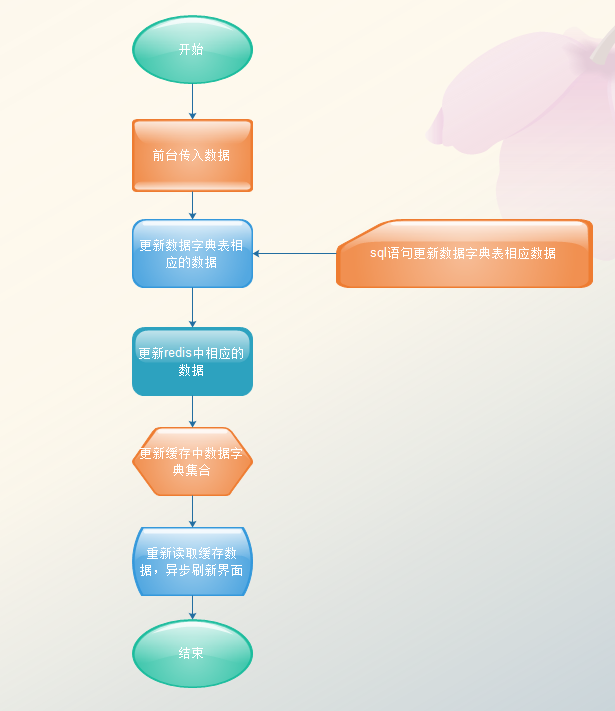
### 新增数据到数据字典表



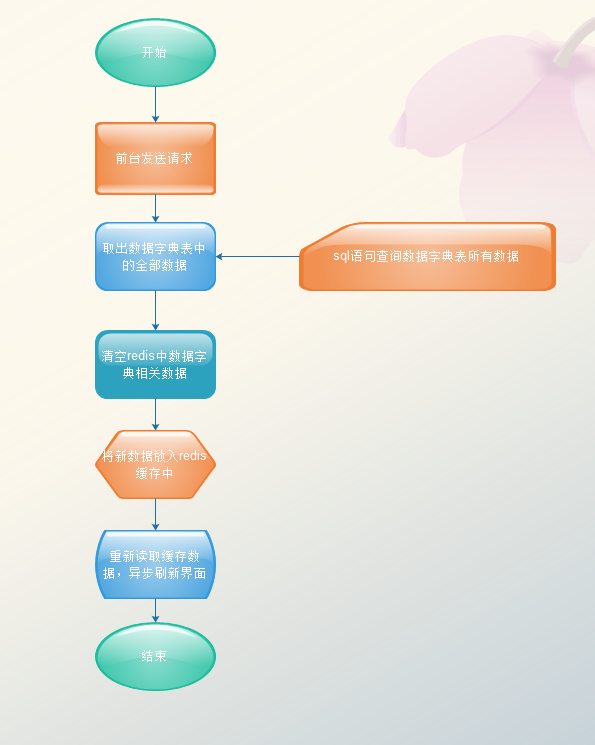
### 删除数据从数据字典表



### 更新数据字典表数据



### 同步数据字典



# Model类设计

## 2.1 model类规划

|  |  |  |
| --- | --- | --- |
| 类名 | 说明 | 描述 |
| ccom.eayun.virtualization.model.BaseDataTree | 数据字典实体类 | 表sys\_data\_tree |
| com.eayun.tag.model.DataTree | 数据字典扩展类 | 表sys\_data\_tree |
|  |  |  |

## 2.2 数据字典表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **数据表名称** | | **sys\_data\_tree** | | | | |
| **功能说明** | | **数据字典表** | | | | |
| **序号** | **字段名称** | **中文名称** | **类型** | **说明** | **非空** | **业务规则** |
| 1 | node\_id | 节点ID | VC(200) |  | N | PK |
| 2 | node\_name | 节点名称 | VC(200) |  |  |  |
| 3 | parent\_id | 父节点ID | VC(50) |  |  |  |
| 4 | sort | 顺序 | decimal (22) |  |  |  |
| 5 | is\_root | 是不是根节点 | VC(50) |  |  |  |
| 6 | memo | 备注 | VC(2000) |  |  |  |
| 7 | flag |  | VC(50) |  |  |  |
| 8 | para1 | 编程参数1 | VC(500) |  |  |  |
| 9 | para2 | 编程参数2 | VC(200) |  |  |  |
| 10 | image\_path | 图片路径 | VC(500) |  |  |  |
| 11 | node\_name\_en | 节点英文名称 | VC(200) |  |  |  |
| 12 | icon | 节点ICON | VC(50) |  |  |  |
| **索引名称** | | | **Unique** | **索引字段** | | |
|  | | |  |  | | |

## redis缓存设计

### 2.3.1通过nodeId得到该条全部的信息

Key格式：sys\_data\_tree:[node\_id]

Value：整个该node\_id对象的Json格式串。

采用Redis K-V实现，如下：

127.0.0.1:6379>set sys\_data\_tree: [node\_id ] dataTreeJSON

dataTreeJSON为sys\_data\_tree 各字段组成的JSON对象。

eg.

set sys\_data\_tree: 0005001 dataTreeJSON

dataTreeJSON:

{

"node\_id": " 0005001 ",

"node\_name": "项目管理 ",

"parent\_id":"0005",

"sort": "2",

"is\_root": "0",

"memo": "项目管理",

"flag":"1",

"para1":"app.clouds.openstack.projectmanage",

"para2":" app.clouds.openstack.projectmanage"

"image\_path":" "

"node\_name\_en":" "

"node\_name\_en":""

}

### 2.3.2通过parentId得到其下所有子集

Key格式：sys\_data\_tree:parent:node\_id:[parent\_id]

Value：该parent对象下的所有node\_id（数据结构是set）

采用Redis的Set实现，如下：

127.0.0.1:6379>sadd sys\_data\_tree:parent:node\_id:[ parent\_id ] [node\_id ]

前缀是sys\_data\_tree:parent:node\_id，之后跟[parent \_id]，即对应一个key是parentId的集合，成员是nodeId。

eg.

sadd sys\_data\_tree:parent:node\_id: 0005 0005001

### 2.3.3通过node\_name\_en得到node\_name

Key格式：sys\_data\_tree:status:vm:[enstatus]，vm表示云主机状态类别，其他诸如disk等。

Value：该类资源英文状态所对应的中文名称。

采用Redis K-V实现，如下：

127.0.0.1:6379>set sys\_data\_tree:status:vm： [ACTIVE] [正常]

sys\_data\_tree:status:vm为说明是云主机的英文状态。

eg.

set sys\_data\_tree:status:vm: ACTIVE 正常