广东移动U-Learning在线平台开发项目

资讯模块 概要设计

**（V 0.1）**

文档修订历史

| 版本号 | 日期 | 撰写人 | 审核人 | 批准人 | 变更摘要 & 修订位置 |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目 录

[1 设计综述 4](#_Toc411007816)

[1.1 设计思路、原则 4](#_Toc411007817)

[1.2 约束、依赖 4](#_Toc411007818)

[2 模块设计 5](#_Toc411007819)

[2.1 综述 5](#_Toc411007820)

[2.2 用例图 5](#_Toc411007821)

[2.3 模块功能 6](#_Toc411007822)

[2.3.1 设置资讯分类 6](#_Toc411007823)

[2.3.2 管理资讯 7](#_Toc411007824)

[2.3.3 推送通知 9](#_Toc411007825)

[2.3.4 订阅资讯 10](#_Toc411007826)

[2.3.5 查看资讯 11](#_Toc411007827)

[2.3.6 查看通知 12](#_Toc411007828)

[3 接口设计 14](#_Toc411007829)

[3.1 接口规范 14](#_Toc411007830)

[3.2 内部接口 14](#_Toc411007831)

[3.2.1 资源辅助信息接口 14](#_Toc411007832)

[3.2.2 资源辅助信息修改接口 15](#_Toc411007833)

[3.3 外部接口 16](#_Toc411007834)

[3.3.1 资源列表接口 16](#_Toc411007835)

[3.3.2 资源明细接口 18](#_Toc411007836)

[3.3.3 订单明细接口 19](#_Toc411007837)

[3.3.4 资源明细修改接口 24](#_Toc411007838)

[3.3.5 订单明细修改接口（待业务层开发） 25](#_Toc411007839)

[3.3.6 资源性能接口 26](#_Toc411007840)

[4 数据架构设计 28](#_Toc411007841)

[4.1 表设计总览 28](#_Toc411007842)

[4.2 详细表设计 28](#_Toc411007843)

[4.2.1 T\_Resource\_OtherInfo：资源辅助信息 28](#_Toc411007844)

[4.3 字段映射说明 29](#_Toc411007845)

# 设计综述

## 设计思路、原则

* 模块开发框架基于系统总体开发框架

## 约束、依赖

1. 暂时以运维人员的角度进行设计，不考虑权限的问题，等开发雏形给用户确认后，再增加权限判断的设计。
2. 资源展示功能，主要是负责资源的展示和数据修改；资源的添加、删除、停用、控制、监控等操作在其他功能模块实现

# 模块设计

## 综述

资讯模块主要是管理员发布资讯、推送通知，学员查看管理员发布的资讯和通知，具体包括如下需求：

1. 管理员设置资讯分类。
2. 管理员对资讯进行增删改查。
3. 管理员向学员推送通知。
4. 学员订阅资讯。
5. 学员查看资讯，可以对资讯进行点赞，收藏，转发操作。
6. 学员查看收到的通知。

## 用例图



根据系统的使用者分类，划分2大类用户：学员和管理员。

|  |  |
| --- | --- |
| **用例名** | **用例描述** |
| 设置资讯分类 | 管理员可以对资讯分类进行增删改查 |
| 管理资讯 | 管理员可以对资讯进行增删改查 |
| 推送通知 | 管理员可以向学员推送通知 |
| 订阅资讯 | 学员可以根据自己的需求订阅自己喜欢的资讯分类，订阅后能查看订阅分类的资讯推荐 |
| 查看资讯 | 学员查看某个资讯的详情，可以进行点赞、收藏、转发操作 |
| 查看通知 | 学员可以查看自己收到的所以通知，并可以查看某个通知详情。 |

## 模块功能

### 设置资讯分类

#### 功能概述

根据当前用户的权限，列出权限范围内的资讯分类。当前设定的权限是：

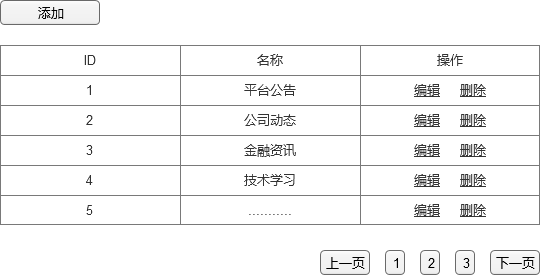
1）管理员可以查看管理所有资讯分类

#### 功能设计

1. **业务方法：**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **接口名** | **方法名** | **方法说明** |
| 1 | 获取资讯分类列表接口 | getInfoClassList() | 按条件分页查询资讯分类列表 |
| 2 | 获取资讯分类详情接口 | getInfoClassByID() | 查看某一个资讯分类详情 |
| 2 | 添加资讯分类接口 | addInfoClass() | 添加资讯分类 |
| 3 | 修改资讯分类接口 | updateInfoClass() | 修改资讯分类 |
| 4 | 删除资讯分类接口 | deleteInfoClass | 删除资讯分类 |

#### 界面设计





设置资讯分类页面包括两大部分，一部分是资讯分类列表页面，一部分是添加资讯分类页面。

* 资讯分类列表页面

展示所有资讯分类，按分页显示。点击添加按钮，跳转到添加资讯分类页面。

* 添加资讯分类页面

输入名称，点击保存按钮，存储到数据库。点击取消按钮，返回资讯分类列表页面。

### 管理资讯

#### 功能概述

管理员对资讯进行增删改查。

#### 功能设计

1. **业务方法：**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **接口名** | **方法名** | **方法说明** |
| 1 | 获取资讯列表接口 | getInfoList() | 按条件分页查询资讯列表 |
| 2 | 获取资讯详情接口 | getInfoByID() | 查看某一个资讯详情 |
| 2 | 添加资讯接口 | addInfo () | 添加资讯 |
| 3 | 修改资讯接口 | updateInfo () | 修改资讯 |
| 4 | 删除资讯接口 | deleteInfo | 删除资讯 |

#### 界面设计





管理资讯包括两个页面，分别如下：

* 资讯列表页面

按资讯分类展示该分类下所有资讯，按分页显示。点击添加按钮，跳转到添加资讯页面。

* 添加资讯页面

选择分类，输入名称、内容，点击保存按钮，存储到数据库。点击取消按钮，返回资讯列表页面。

### 推送通知

#### 功能概述

管理员可以向指定组织、职位、学员推送通知。

#### 功能设计

1. **业务方法：**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **接口名** | **方法名** | **方法说明** |
| 1 | 推送通知接口 | pushNotification () | 推送通知 |

#### 界面设计



* 推送通知页面

选择推送对象及对应的对象值，选择重要性，输入标题，输入内容，点击推送按钮，通知推送。点击取消按钮返回。

### 订阅资讯

#### 功能概述

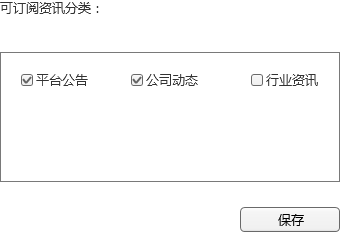
学院订阅资讯分类

#### 功能设计

1. **业务方法：**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **接口名** | **方法名** | **方法说明** |
| 1 | 获取资讯分类列表接口 | getInfoClassList () | 获取可订阅资讯分类列表 |
| 2 | 获取已订阅资讯分类列表接口 | getSubscribeInfoClassList() | 获取学员已订阅分类 |
| 3 | 订阅资讯分类接口 | subscribeInfoClass () | 订阅资讯分类 |

#### 界面设计



* 资讯分类订阅页面

选择要订阅分类，点击保存按钮，存储数据库。

### 查看资讯

#### 功能概述

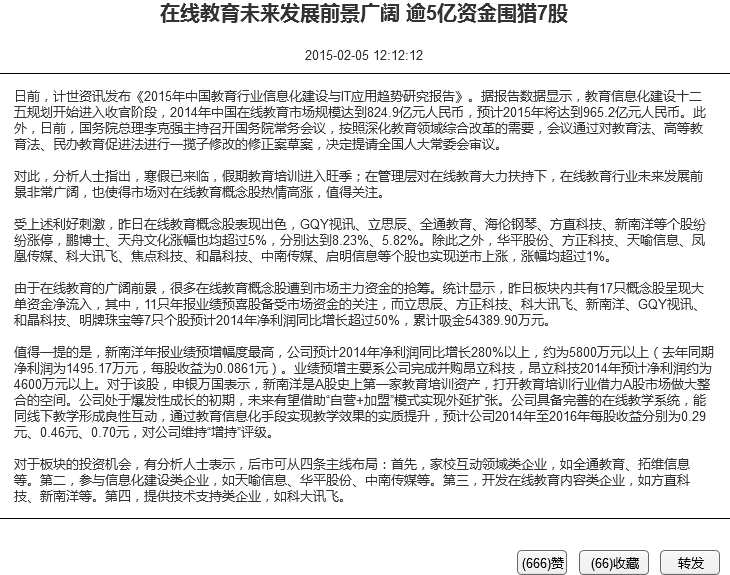
学员查看某一资讯详情，可以点赞、收藏、转发。

#### 功能设计

1. **业务方法：**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **接口名** | **方法名** | **方法说明** |
| 1 | 获取资讯接口 | getInfoById () | 获取资讯详情 |
| 2 | 点赞接口 | goodInfo() | 学员对资讯点赞 |
| 3 | 收藏接口 | collectInfo () | 学员收藏资讯 |
| 3 | 转发接口 | forwardInfo () | 学员转发资讯 |

#### 界面设计



* 查看资讯页面

展示资讯详情，学员可以点击赞按钮，收藏按钮，转发按钮。

### 查看通知

#### 功能概述

学员查看收到通知

#### 功能设计

1. **业务方法：**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **接口名** | **方法名** | **方法说明** |
| 1 | 获取收到通知列表接口 | getNotificationListByUID () | 获取学员收到的所有通知 |
| 2 | 获取通知详情接口 | getNotificationByID() | 获取某一个通知详情 |

#### 界面设计





* 通知列表页面

分页展示学员收到的通知，按重要性、时间排序。

* 通知详情页面

展示通知详情

# 接口设计

## 接口规范

接口采用Restful Web Service技术。

数据传输采用JSON数据格式。

调用Cloud 2.0的接口URL前面统一为：http://XXX/ ws/outer/，其中XXX是Cloud 2.0的WS服务器地址，后面是接口路径。一个完整的接口URL是该通用地址加上每个接口方法即可，如创建订单接口的接口方法是createorder，则访问该接口的URL为http://XXX/ ws/outer/createorder/，使用JSON数据格式通过POST方式传递数据，Cloud 2.0即可接收到创建订单的数据。

## 内部接口

### 资源辅助信息接口

#### 接口说明

调用发起方： 资源展示模块

调用接收方： 资源展示模块

接口功能说明：

获取资源相关的辅助信息。

#### 接口定义

1. **接口调用方法**

getResourceOtherInfo

1. **Request URL**

|  |
| --- |
| /rest/resource/getResourceOtherInfo/｛resourceId｝/｛domainName｝ |

1. **接口数据格式**

采用Json数据格式，使用POST传输

1. **接口输入参数**
2. resourceId：资源ID
3. domainName：域名
4. **接口返回参数：**

Json数据，类似订单明细接口

### 资源辅助信息修改接口

#### 接口说明

调用发起方： 资源展示模块

调用接收方： 资源展示模块

接口功能说明：

修改资源相关的辅助信息。

#### 接口定义

1. **接口调用方法**

updateResourceOtherInfo

1. **Request URL**

|  |
| --- |
| /rest/resource/updateResourceOtherInfo/｛resourceId｝/｛domainName｝ |

1. **接口数据格式**

采用Json数据格式，使用POST传输

1. **接口输入参数**
2. resourceId：资源ID
3. domainName：域名
4. 资源辅助信息（ResourceOtherInfo节点）：

包括监控状态说明、立项号、应用程序类型、固定停机窗口、操作前联系、操作后联系、备注等。

1. **接口返回参数：**

|  |  |  |
| --- | --- | --- |
| **返回值** | **类型** | **说明** |
| {Result:1;errList:} | JSON | 成功 |
| {Result:0;errList:错误列表} | JSON | 失败 |

输出为一个JSON对象，格式为“result:结果;errList:错误描述”。如果成功：“结果”为1，“错误描述”为空串；如果失败，“结果”为空串，“错误列表”为错误描述。约定“result:”和“errList:”不管成功与否都必须有，只是值可以为空串。

## 外部接口

### 资源列表接口

#### 接口说明

调用发起方： 资源展示模块

调用接收方： 资源层模块

接口功能说明：

按条件分页查询资源信息列表。

#### 接口定义

1. **接口调用方法**

queryHibernateDomainList，资源层模块提供

1. **接口来源文档**

《Cloud2.0\_ Iaas平台资源管理接口1.0》

1. **接口数据格式**

采用Json数据格式，使用POST传输

1. **接口输入参数**

1、domainName：域名

虚拟机：TBGuest

2、condtion：查询条件

属于资源层的字段，可以直接作为条件传入。

属于其他层的字段，需要先通过条件查询获取resource id，然后再以此作为条件传入。（如CMDBID：可以到t\_order\_item表中，通过code过滤记录，得到resource id）

注：

* 1. resource\_id，关联的是tb\_host或tb\_guest的ID。可以通过前缀来区分， 如： bare\_id，vm\_id。
  2. 对于 宿主机的信息，需要先用queryHibernateDomainList接口，到TB\_HOST域中，查询hostType=’ hypervisor’的hostID，然后作为条件传入此接口
  3. 对于 宿主机控制器的信息，需要先用queryHibernateDomainList接口，到TB\_Controller域中查询ControlerID，然后作为条件传入此接口

3、start：开始记录条数：分页查询用

4、end：结束记录条数：分页查询用

5、order：排序

1. **接口返回参数**

PagedQueryResult （包含hiberante对应的域对象列表）

#### 接口样例

1. Sample URL:

http://16.157.133.97:8080/vresman/vresquery/hibernate/TBGuest/null/0/1/id%20asc

1. Sample数据

{"records":[{"id":2,"hostId":2,"hostName":"hair\_1403256672174","desc":"auto booked ","localId":"87f99693-e3b5-4e44-9002-e4e1922c596c","ip":"11.11.11.3","netMask":"255.255.255.0","gateway":"11.11.11.1","dns":"8.8.8.8,11.11.11.1","vlan":null,"diskNum":1,"diskVolume":19,"osType":"redhat\_64bit","imageId":3,"user":null,"password":null,"arch":"x86","socketNum":1,"coreNum":1,"threadNum":1,"memory":2096,"status":9,"monitorStatus":-1,"ctime":1403299872000,"mtime":1404760150000}],"start":0,"end":1,"errorCode":0,"hintMessage":null}

### 资源明细接口

#### 接口说明

调用发起方： 资源展示模块

调用接收方： 资源层模块

接口功能说明：

获取资源的详细信息。

#### 接口定义

1. **接口调用方法**

getHibernateDomainInfo，资源层模块提供

1. **接口来源文档**

《Cloud2.0\_ Iaas平台资源管理接口1.0》

1. **接口数据格式**

采用Json数据格式，使用POST传输

1. **接口输入参数**

1、domainName：域名

虚拟机：TBGuest

宿主机：TBHost

宿主机控制器：TBController

2、domainId：主键

转入资源数据ID

1. **接口返回参数**

Json数据（包含hiberante对应的域对象明细数据）

#### 接口样例

1. Sample URL:

<http://16.157.133.97:8080/vresman/vresquery/hibernate/detail/TBGuest/2>

1. Sample数据

{"id":2,"hostId":2,"hostName":"hair\_1403256672174","desc":"auto booked ","localId":"87f99693-e3b5-4e44-9002-e4e1922c596c","ip":"11.11.11.3","netMask":"255.255.255.0","gateway":"11.11.11.1","dns":"8.8.8.8,11.11.11.1","vlan":null,"diskNum":1,"diskVolume":19,"osType":"redhat\_64bit","imageId":3,"user":null,"password":null,"arch":"x86","socketNum":1,"coreNum":1,"threadNum":1,"memory":2096,"status":9,"monitorStatus":-1,"ctime":1403299872000,"mtime":1404942263000}

### 订单明细接口

#### 接口说明

调用发起方： 资源展示模块

调用接收方： 业务层模块

接口功能说明：

获取资源相关的订单详细信息。

#### 接口定义

1. **接口调用方法**

getOrderById，业务层模块提供

1. **接口来源文档**

《海尔项目开发接口文档v1.1》

1. **接口数据格式**

采用Json数据格式，使用POST传输

1. **接口输入参数**
2. orderId：订单ID

注：通过Resource ID到T\_order\_item表中关联找出所对应的Order\_ID，然后再作为参数找出订单相关的明细。

1. **接口返回参数**

Json数据

|  |  |
| --- | --- |
| 对应信息 | 对应节点 |
| 用户信息 | user节点 |
| 订单信息 | 根节点 |
| 项目信息 | project节点 |

#### 接口样例

1. Sample URL:

http://localhost:8080/cloud2.0/rest/order/getOrderById/1

1. Sample数据

|  |
| --- |
| {  "orderId": "1",  "orderNO": "PSI-2014-mail-1001",  "status": "evaluated",  "remark": "nothing",  "changeReason": "Server system evaluation",  "newFlag": "project",  "createDate": "2014-03-07",  "lastUpdateDate": "2014-09-07",  "periodStartDate": "2014-03-07",  "periodEndDate": "2014-03-07",  "orderNum": "psi-jxjg-hp-20140101-51",  "oldCode": null,  "sendUser": "unsent",  "user": {  "id": "2",  "userName": "李四",  "type": "1",  "title": "1",  "email": "admin@hp.com",  "phone": "13613613610",  "bizName": "小翟",  "bizPhone": "13613613610",  "bizEmail": "admin@hp.com",  "org": {  "code": "10001",  "id": "1",  "name": "psi",  "type": "A股",  "status": "1",  "buFu": "PSI",  "bizDept": "FU投发HIG流程信息经营体"  }  },  "project": {  "sn": null,  "id": "1",  "name": "邮件系统",  "appProvider": "小王",  "appPerson": "小李",  "appPersonTel": "13813813810",  "appPersonEmail": "li@163.com",  "psiInterface": "FU投发HIG流程信息经营体",  "psiSysAdmin": "manager",  "psiSysAdminTel": "13813813822",  "psiSysAdminEmail": "m1@163.com",  "status": "1",  "remark": "stability",  "createDate": "2014-03-06",  "creator": "admin",  "updateDate": "2014-03-08",  "updateMan": "admin1"  },  "detailItemVOList": [  {  "itemId": "1",  "name": "Linux",  "code": "DB10001",  "type": "PM",  "ip": "129.0.0.3",  "price": "100.0",  "feestartDate": "2014-03-19",  "status": "deploy",  "workStatus": 0,  "configItems": [  {  "itemName": "CPU",  "itemValue": "1C",  "itemId": "1"  },  {  "itemName": "内存",  "itemValue": "2G",  "itemId": "2"  },  {  "itemName": "硬盘类型",  "itemValue": "100G",  "itemId": "3"  },  {  "itemName": "数据库",  "itemValue": "Mysql",  "itemId": "4"  },  {  "itemName": "操作系统",  "itemValue": "Windows",  "itemId": "5"  },  {  "itemName": "外网IP",  "itemValue": "129.0.0.0",  "itemId": "6"  }  ]  }  ],  "orderApprovalDetailVoList": [  {  "approveNo": 20140515,  "approver": "张三",  "result": "resiltList",  "approveDesc": "approveDesc1",  "approveTime": "2014-05-15",  "remark": "remarkNothing",  "creator": "小张",  "createDate": "2014-05-15"  }  ]  } |

### 资源明细修改接口

#### 接口说明

调用发起方： 资源展示模块

调用接收方： 资源层模块

接口功能说明：

修改资源相关的详细信息。

#### 接口定义

1. **接口调用方法**

updateHibernateDomain，资源层模块提供

1. **接口来源文档**

《Cloud2.0\_ Iaas平台资源管理接口1.0》

1. **接口数据格式**

采用Json数据格式，使用POST传输

1. **接口输入参数**

1、domainName：域名

虚拟机：TBGuest

2、updateFields ：更新的字段

数据传入使用MAP的形式，字段名称-新值

3、condition：限定条件

可以传入资源ID进行数据限定

1. **接口返回参数**

更新个数，如果成功更新的条数为1，则表示成功，如果成功更新的条数为0，则表示失败。

### 订单明细修改接口（待业务层开发）

#### 接口说明

调用发起方： 资源展示模块

调用接收方： 业务层模块

接口功能说明：

修改资源相关的订单详细信息。

#### 接口定义

1. **接口调用方法**

updateOrder，业务层模块提供

1. **接口来源文档**

《Cloud2.0\_ Iaas平台资源管理接口1.0》

1. **接口数据格式**

采用Json数据格式，使用POST传输

1. **接口输入参数**

1、orderId：所需更新订单的主键

传入资源相关的订单ID

注：通过Resource ID到T\_order\_item表中关联找出所对应的Order\_ID，然后再作为参数输入。

2、**用户参数（user节点）：**

包括申请BU/FU、申请部门、应用用户、应用用户邮箱、应用用户电话、PSI经营体团队长、团队长邮箱、团队长电话、经营体长等。

3、**项目参数（project节点）：**

包括项目名称、PSI系统负责人、PSI系统负责人邮箱、PSI系统负责人电话、PSI部门 、应用开发支持责任人、应用开发支持责任人邮箱、应用开发支持责任人电话等。

4、**订单参数（根节点）：**

包括重要级别等

1. **接口返回参数**

|  |  |  |
| --- | --- | --- |
| **返回值** | **类型** | **说明** |
| {Result:1;errList:} | JSON | 成功 |
| {Result:0;errList:错误列表} | JSON | 失败 |

输出为一个JSON对象，格式为“result:结果;errList:错误描述”。如果成功：“结果”为1，“错误描述”为空串；如果失败，“结果”为空串，“错误列表”为错误描述。约定“result:”和“errList:”不管成功与否都必须有，只是值可以为空串。

### 资源性能接口

#### 接口说明

调用发起方： 资源展示模块

调用接收方： 监控模块

接口功能说明：

获取资源的性能监控信息。

#### 接口调用说明

1.      先将这个包放入工程中，cloud2.0之前应该有，所以覆盖即可。

2.       监控包其实也是单独的一套独立环境，所以在使用里面的接口之前需要预先初始化和销毁2个动作，可以认为是spring的初始化和销毁动作，如果只是很少调用可以将初始化/销毁的方法写在自己的类中，如果是要经常使用的建议添加到系统初始化的事件中，比如ServletContextListener。cloud2.0已经在定时调度中做了初始化，不过由于会经常使用这些接口，所以还是挪到工程级的事件中。一个工程最好只初始化一份监控，初始化和销毁的方法如下：

EnvironmentFactory.getInstance().initEnvironment();

EnvironmentFactory.getInstance().destroyEnvironment();

附件中有一个其他工程的初始化例子，可以参考。

注意：你在ServletContextListener中添加了的话最好把以前代码中的初始化/销毁的2个方法删掉，保证上面的2个方法只在你的工程中出现1次。

3.       初始化/销毁的代码完成后，就可以使用接口了，需要的3个接口如下：

PerformanceDataService. getCPUPercentByHost (String hostName, RRAResolutionType resolutionType, Date startDate, Date endDate);

PerformanceDataService. getMemoryPercentByHost (String hostName, RRAResolutionType resolutionType, Date startDate, Date endDate);

PerformanceDataService. getDiskPercentByHost (String hostName, RRAResolutionType resolutionType, Date startDate, Date endDate);

调用方法可以参考PerformanceDataServiceImplTest. testGetMemWeeklyPerformanceDataByHost 的例子

4.       参数中的RRAResolutionType含义如下：

这个枚举包含3种值，是监控目前具有的3种采集数据

AVG\_WEEKLY("AVERAGE", 300)：平均值一周数据，精度300秒

AVG\_QUARTER("AVERAGE", 1800)：平均值90天数据，精度1800秒

MAX\_QUARTER("MAX", 86400)：最大值90天数据，精度86400秒

精度是指点与点之间的间隔时间，只要一周的数据可以使用AVG\_WEEKLY或AVG\_QUARTER都可以。

#### 接口调用示例

前台js示例参考：

initLineData : **function**(e) {

**var** type="line";

**var** config = {

                        url : commons.webserversite+'/am\_lineWithDurationOnValueAxis.json', //'/rest/amcharts/charts/'+type,

                        type : 'GET',

                        success : 'indexPage.initLineDataSuccess'

                   };

              doAJax.doConfig(config);

          },

          initLineDataSuccess : **function**(chartData) {

              AmCharts.ready(amCharts.chartSmoothLine("chartLinediv", chartData,"time", "百分比"));

          },

# 数据架构设计

## 表设计总览



## 详细表设计

### T\_Resource\_OtherInfo：资源辅助信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Code** | **Data Type** | **Primary** | **Foreign Key** | **说明** |
| ID | id | varchar(60) | TRUE | FALSE |  |
| 资源ID | RESOURCE\_ID | varchar(60) | FALSE | FALSE |  |
| 域名 | Domain\_Name | varchar(50) | FALSE | FALSE |  |
| 监控状态说明 | MONITER\_STATUS\_MEMO | varchar(2000) | FALSE | FALSE |  |
| 固定停机窗口 | FIXED\_STOP\_WINDOW | varchar(2000) | FALSE | FALSE |  |
| 操作前联系 | CONTACT\_BEFORE\_OPERATION | varchar(2000) | FALSE | FALSE |  |
| 操作后联系 | CONTACT\_AFTER\_OPERATION | varchar(2000) | FALSE | FALSE |  |
| 立项号 | PROJECT\_CODE | varchar(100) | FALSE | FALSE |  |
| 应用程序类型 | APPLICATION\_TYPE | varchar(100) | FALSE | FALSE |  |
| 备注 | REMARK | varchar(4000) | FALSE | FALSE |  |
| 创建日期 | create\_date | date | FALSE | FALSE |  |
| 创建人 | creator | varchar(50) | FALSE | FALSE |  |
| 修改日期 | update\_date | date | FALSE | FALSE |  |
| 修改人 | update\_man | varchar(50) | FALSE | FALSE |  |

## 字段映射说明

由于资源展示所用到的字段牵涉面比较广，所以需要在开发前详细了解它们之间的关系，具体请参见《资源展示字段说明表.xls》

该文件中会包含对访问权限的定义（目前此项待定）

需要特别注意的是，字段来源总共分为3大类：资源、业务、辅助、其他。

**资源**：需要通过资源层提供的接口去操作，这类数据主要集中在资源信息、宿主机信息、宿主机控制器信息。

**业务**：需要通过业务层提供的接口去操作，这类数据包括：订单信息、用户信息、部门信息、项目信息。

**辅助**：直接通过访问数据库去操作，这类数据是用于辅助管理目的而产生的，属于其他信息。

**其他**：这类数据都是系统自动计算生成的，在数据库中没有保存记录。