**说明**

一次同步一共需要发送两次请求，**第一次请求**作用为获取服务器端对数据库的修改，**第二次请求**作用为向服务器端提交对数据库的修改。

由于两端的数据库互相依赖，形成了双向数据流，又由于同步时间是有延迟的，会导致丢失修改等问题，所以web端提交的新增、修改、删除等操作都不直接对服务器端数据库操作，而是将其缓存在内存之中，在**第一次请求**的时候，将其作为响应内容发送至PC端，PC端将其与本地缓存（PC端所提交的CUD操作可以直接存入本地数据库，但也要缓存起来）对比之后，再将符合要求的请求存入数据库，同时，也对缓存内容进行校正与整合，在**第二次请求**的时候，再作为请求体提交至服务器，更新服务器端的数据库。

**第一次请求**

请求地址：http://[域名]/ extractCache/[区域编号]

请求方式：GET

获取到的基本数据结构说明：

nodeInfo：对node\_info表的操作

set：数组，对表的更新操作，元素为对象

nodeId：要修改的节点ID

content：修改的具体内容

time：修改的提交时间，是一个时间戳

del：数组，对表的删除操作，元素为要删除的节点ID

userInfo：对user\_info表的操作

set：数组，对表的更新操作，元素为对象

cardNo：要修改的用户卡号

content：修改的具体内容

time：修改的提交时间，是一个时间戳

del：数组，对表的删除操作，元素为要删除的用户卡号

userPermission：对user\_permission表的操作

add：数组，对表的插入操作，元素为对象，对象内容为记录个各个字段值

del：数组，对表的删除操作，元素为要删除的记录id

**注意：**在userPermission的add数组中的元素

startPeriod和endPeriod格式为yyyy-mm-dd的字符串，如："2012-03-19"

startTime和endTime格式为hh:mm的字符串，如："13:00"，"19:30"

示例：

{

"nodeInfo": {

"set": [

{

"nodeId": 11,

"content": { "nodeName": "音乐室" },

"time": 1481610287376

},{

"nodeId": 14,

"content": { "nodeName: "杂物间" },

"time": 1481610287934

}

],

"del": [15, 18, 20]

},

"userInfo": {

"set": [

{

"cardNo": 1717,

"content": {

"personId": "itt13021",

"name": "撒网行",

"phone": 2141123412

},

"time": 1481610245635

}

],

"del": [17, 31]

},

"userPermission": {

"add": [

{

"cardNo": 12312，

"personId": "itt13038",

"name": "捱三顶",

"nodeId": 31,

"type": 0,

"startPeriod": "2012-03-04",

"endPeriod": "2013-09-10",

"startTime": "13:21",

"endTime": "19:32"

}

],

"del": [6, 20]

}

}

**注意：**若是数组内没有内容，该数组还是会存在，只不过作为空数组，切勿将其移除

**第二次请求**

请求地址：http://[域名]/commit/[区域编号]

请求方式：POST

发送的数据结构说明：

基于第一次请求收到的对象进行扩展（用同一个对象就好），如果node\_info、user\_info表有插入请求，请在相应的对象上增加类型为数组的add属性，并将要插入的记录变为对象置入add数组中，示例：

……

"nodeInfo": {

"set": [...],

"del": [...],

"add": [

{

"nodeId": 321,

"place": 3,

"name": "巴拉巴拉",

"status": 0,

"time": "2016-02-29 13:00"

}

]

},

……

**注意：**这里的time字段并无限制，web端存储的内容为String，PC端上传的是什么，存储的内容就是什么

同样的，若是user\_permission表有表的更新操作，请在相应的对象下添加类型为数组的set属性，每一个修改项作为一个元素，需要有以下两个个字段：

id // 要修改的权限id

content // 要修改的具体内容

示例：

……

"userPermission": {

"del": [...],

"add": [...],

"set": [

{

"id": 12,

"content": { "type": 3 }

}, {

"id": 33,

"content": { "persionId": 'itt13323' }

}, {

"id": 28,

"content": {

"nodeId": 342,

"startPeriod": 'dwqsa',

"endTime": 'asda213'

}

}

]

},

……

**注意：** 这里的startPeriod, endPeriod, startTime, endTime字段同样也无限制，上传的是什么，存储的就是什么，有格式限制的是第一次请求时获取到的内容

最后再将扩展完成的对象作为请求体传输就行，不过需要设置一下请求头，将Content-Type字段的值设置为application/json，否则服务器端解析会出错。