Management Center Server API设计

版本号：V1.0

日期：2017.6.1

修订记录:

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 作者 | 修改描述 |
| V1.0 | 2017.6.1 | 陈晨 | 初稿 |

# 基础信息模块

## 楼栋信息

### 楼栋信息录入

* URL：POST /api/addBuilding
* 参数：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| building | YES | 楼栋号 |

* 示例：

{

"building": 2

}

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | Int | 状态编码1:成功 |
| ret | JSON | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": "success"

}

* + **失败：**

{

"state": 0,

"ret": "error"

}

### 楼栋信息查询

* URL：GET /api/getBuildings
* 参数：String

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| page | YES | 页 |
| rpp | YES | 每页多少条数据 |

* 返回：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度（字节） | 说明 |
| state | Int | 1 | 状态编码1:成功 |
| ret | JSON | 40 | 返回结果 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ret** | | | |
| 字段名 | 数据类型 | 长度（字节） | 说明 |
| building | String | 40 | 楼栋信息 |
| x | int | 2 | 横坐标 |
| y | int | 2 | 纵坐标 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": [

{

"id": "1",

"building": "1",

"x": "1",

"y": "1"

},

{

"id": "2",

"building": "2",

"x": "1",

"y": "1"

}

],

"total": 2

}

* + **失败：**

{

"state": 0,

"ret": ""

}

### 楼栋信息修改

* URL：PUT /api/editBuilding
* 参数：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| id | YES | 楼栋ID |
| building | YES | 楼栋信息 |
| x | int | 横坐标 |
| y | int | 纵坐标 |

* 示例：

{

"id" : "8",

"name": "1201",

"x": "1201",

"y": "1201"

}

* 返回：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度（字节） | 说明 |
| state | Int | 1 | 状态编码1：成功,2：失败 |
| ret | String | 40 | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败：**

{

"state": 0,

"ret": ""

}

### 删除楼栋信息

* URL：DEL /api /removeBuilding
* 参数：String

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| id\* | YES | 楼栋ID |

* 返回：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度（字节） | 说明 |
| state | Int | 1 | 状态编码1:成功 |
| ret | String | 40 | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败：**

{

"state": 0,

"ret": ""

}

## 单元信息

### 单元信息录入

* URL：POST /api/ addUnit
* 参数：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| building | YES | 楼栋ID |
| unit | YES | 单元名称 |

* 示例：

{

"building": 2,

"unit": "2"

}

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | Int | 状态编码1:成功 |
| ret | JSON | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": "success"

}

* + **失败：**

{

"state": 0,

"ret": "error"

}

### 单元信息查询

* URL：GET /api/getUnits
* 参数：String

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| building | NO | 楼栋ID |
| page | NO | 页 |
| rpp | NO | 每页多少条数据 |

* 返回：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度（字节） | 说明 |
| state | Int | 1 | 状态编码1:成功 |
| ret | JSON | 40 | 返回结果 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ret** | | | |
| 字段名 | 数据类型 | 长度（字节） | 说明 |
| id | Int | 40 | 单元ID |
| buildingId | int | 2 | 楼栋ID |
| building | STRING | 2 | 楼栋名称 |
| unit | STRING | 3 | 单元名称 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": [

{

"id": "1",

"unit": "01",

"buildingId": "1",

"building": "1"

}

],

"total": 1

}

* + **失败：**

{

"state": 0,

"ret": ""

}

### 单元信息修改

* URL：PUT /api/editUnit
* 参数：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| id | YES | 单元ID |
| building | YES | 楼栋ID |
| unit | YES | 单元名称 |

* 示例：

{

"id" : "8",

"building ": "1201",

"unit ": "1201"

}

* 返回：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度（字节） | 说明 |
| state | Int | 1 | 状态编码1：成功,2：失败 |
| ret | String | 40 | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败：**

{

"state": 0,

"ret": ""

}

### 删除单元信息

* URL：DEL /api / removeUnit
* 参数：String

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| id\* | YES | 单元ID |

* 返回：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度（字节） | 说明 |
| state | Int | 1 | 状态编码1:成功 |
| ret | String | 40 | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败：**

{

"state": 0,

"ret": ""

}

## 房间信息

### 房间信息录入

* URL：POST /api/addRoom
* 参数：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| building | YES | 楼栋ID |
| unit | YES | 单元ID |
| room | YES | 房间号 |

* 示例：

{

"building": 2,

"unit": "2",

"room": 2,

}

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | Int | 状态编码1:成功 |
| ret | JSON | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": "success"

}

* + **失败：**

{

"state": 0,

"ret": "error"

}

### 房间信息查询

* URL：GET /api/getRooms
* 参数：String

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| building | NO | 楼栋ID |
| unit | NO | 单元ID |
| page | NO | 页 |
| rpp | NO | 每页多少条数据 |

* 返回：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度（字节） | 说明 |
| state | Int | 1 | 状态编码1:成功 |
| ret | JSON | 40 | 返回结果 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ret** | | | |
| 字段名 | 数据类型 | 长度（字节） | 说明 |
| id | Int | 40 | 单元ID |
| buildingId | int | 2 | 楼栋ID |
| building | STRING | 2 | 楼栋名称 |
| unitId | INT | 32 | 单元ID |
| unit | STRING | 3 | 单元名称 |
| room | STRING | 5 | 房间名称 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": [

{

"id": "1",

"unitId": "1",

"buildingId": "1",

"building": "1",

"unit": "01",

"room": "0101"

},

{

"id": "2",

"unitId": "1",

"buildingId": "1",

"building": "1",

"unit": "01",

"room": "0102"

},

{

"id": "3",

"unitId": "1",

"buildingId": "1",

"building": "1",

"unit": "01",

"room": "0201"

},

{

"id": "4",

"unitId": "1",

"buildingId": "1",

"building": "1",

"unit": "01",

"room": "0202"

},

{

"id": "5",

"unitId": "1",

"buildingId": "1",

"building": "1",

"unit": "01",

"room": "0301"

},

{

"id": "6",

"unitId": "1",

"buildingId": "1",

"building": "1",

"unit": "01",

"room": "0302"

},

{

"id": "7",

"unitId": "1",

"buildingId": "1",

"building": "1",

"unit": "01",

"room": "0401"

},

{

"id": "8",

"unitId": "1",

"buildingId": "1",

"building": "1",

"unit": "01",

"room": "0402"

},

{

"id": "9",

"unitId": "1",

"buildingId": "1",

"building": "1",

"unit": "01",

"room": "0501"

},

{

"id": "10",

"unitId": "1",

"buildingId": "1",

"building": "1",

"unit": "01",

"room": "0502"

}

],

"confirm": "Are you sure remove this?",

"total": 20

}

* + **失败：**

{

"state": 0,

"ret": ""

}

### 单元信息修改

* URL：PUT /api/editRoom
* 参数：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| id | YES | 单元ID |
| building | YES | 楼栋ID |
| unit | YES | 单元ID |
| room | YES | 房间号 |

* 示例：

{

"id" : "8",

"building ": "1201",

"unit ": "1201",

"room ": "1201"

}

* 返回：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度（字节） | 说明 |
| state | Int | 1 | 状态编码1：成功,2：失败 |
| ret | String | 40 | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败：**

{

"state": 0,

"ret": ""

}

### 删除房间信息

* URL：DEL /api /removeRoom
* 参数：String

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| id\* | YES | 单元ID |

* 返回：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度（字节） | 说明 |
| state | Int | 1 | 状态编码1:成功 |
| ret | String | 40 | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败：**

{

"state": 0,

"ret": ""

}

## 设备分类信息

### 设备信息分类列表

* URL：GET /api/deviceCate
* 参数

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| name | NO | 类别名称 |
| page | NO | 页数 |
| rpp | NO | 每页显示条数 |

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | String | 状态1:成功 |
| ret | JSON | 返回结果 |

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| id | Int |  |
| name | String | 分类名称 |
| models | JSON | 设备型号列表 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": [

{

"id": "1",

"name": "01",

"models": [

"ADT-10",

"DMT-100"

]

},

{

"id": "1",

"name": "02",

"models": [

"Guard-100",

"GAT-10"

]

}

]

}

* + **失败**

{

"state": 0,

"ret": ""

}

### 添加设备信息分类

* URL：POST /api/deviceCate
* 参数：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| name | YES | 分类名称 |
| models | YES | 设备型号 |

* 示例：

{

"name": "01",

"models": "ADT-10"

}

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | String | 状态1：成功，0：失败 |
| ret | String | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败**

{

"state": 0,

"ret": ""

}

### 删除设备分类

* URL：DEL /api/deviceCate
* 参数：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| id | YES | 分类ID |

* 返回：JSON

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度（字节） | 说明 |
| state | String | 40 | 状态1:成功， 0：失败 |
| ret | String |  | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败**

{

"state": 0,

"ret": ""

}

### 修改设备信息分类

* URL：PUT /api/deviceCate
* 参数：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| id | YES | 分类ID |
| name | YES | 分类名称 |
| model | YES | 设备型号 |
| remark | NO | 备注 |

* 示例：

{

"id":"8",

"name": "01",

"model": "ADT-10",

"remark":"新品"

}

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | String | 状态1:成功 |
| ret | String | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败**

{

"state": 0,

"ret": ""

}

## 业主信息

### 添加业主信息

* URL：POST /api/owner
* 参数：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| name | YES | 业主姓名 |
| avator | NO | 业主头像 |
| building | YES | 栋 |
| block | YES | 单元 |
| room | YES | 室 |
| phone\_primary | NO | 联系方式1 |
| phone\_secondary | NO | 联系方式2 |
| remark | NO | 用户描述 |

* 示例：

{

"name":"高欣",

"avator":"/img/6002.png",

"building":"20",

"block":"23",

"room":"203",

"phone\_primary":"13512000000",

"phone\_secondary":"13512000000",

"remark":"描述描述"

}

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | String | 状态1：成功，0：失败 |
| ret | String | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败**

{

"state": 0,

"ret": ""

}

### 修改业主信息

* URL：PUT /api/owner
* 参数：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| id | YES | 业主id |
| name | YES | 业主姓名 |
| avator | NO | 业主头像 |
| building | YES | 栋 |
| block | YES | 单元 |
| room | YES | 室 |
| phone\_primary | NO | 联系方式1 |
| phone\_secondary | NO | 联系方式2 |
| remark | NO | 用户描述 |

* 示例：

{

"id":"2",

"name":"高欣",

"avator":"/img/6002.png",

"building":"20",

"block":"23",

"room":"203",

"phone\_primary":"13512000000",

"phone\_secondary":"13512000000",

"remark":"描述描述"

}

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | String | 状态1：成功，0：失败 |
| ret | String | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败**

{

"state": 0,

"ret": ""

}

### 删除业主信息

* URL：DEL /api/Owner
* 参数：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| id | YES | 业主id |

* 示例：

{

"id":"2",

}

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | String | 状态1：成功，0：失败 |
| ret | String | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败**

{

"state": 0,

"ret": ""

}

### 查询业主信息

* URL：GET /api/Owner
* 参数：String

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| id | YES | 业主ID |
| name | NO | 业主姓名 |
| building | NO | 栋 |
| block | NO | 单元 |
| room | NO | 室 |
| page | YES | 页数 |
| rpp | YES | 每页显示条数 |

* 返回：JSON

|  |  |
| --- | --- |
| 字段名 | 说明 |
| state | 状态1:成功 |
| ret | 返回结果 |

* 示例：

{

"state": 1,

"ret": [

{

"id": "1",

"name": "高欣",

"avator": "/img/6002.png",

"building": "20",

"block": "23",

"room": "203",

"phone\_primary": "13512000000",

"phone\_secondary": "13512000000",

"remark": "描述描述"

},

{

"id": "2",

"name": "大脸猫",

"avator": "/img/6003.png",

"building": "20",

"block": "23",

"room": "203",

"phone\_primary": "13512000000",

"phone\_secondary": "13512000000",

"remark": "我是大脸猫"

},

{

"id": "3",

"name": "大圣",

"avator": "/img/6003.png",

"building": "20",

"block": "23",

"room": "203",

"phone\_primary": "13512000000",

"phone\_secondary": "13512000000",

"remark": "我是大圣"

}

],

"totalNum": 234

}

## 室内机

### 室内机录入

* URL：POST /api /indoor
* 参数：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| building | YES | 栋 |
| block | YES | 单元 |
| room | YES | 室 |
| ip | YES | IP |
| sip | YES | Sip帐号 |
| model | YES | 型号 |
| remark | NO | 备注 |

* 示例：

{

"building": "3",

"block": "2",

"room": "1010",

"ip": "192.168.1.100",

"sip":"8008",

"model": "DMT-100",

"remark": "新款"

}

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | Int | 状态编码1:成功,0:失败 |
| ret | String | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": "ok"

}

* + **失败：**

{

"state": 0,

"ret": ""

}

### 室内机查询

* URL：GET /api/indoor/ building=101&block=1&room=1001&ip=192.168.1.100&page=2&rpp=20
* 参数：String

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| building | NO | 栋 |
| block | NO | 单元 |
| room | NO | 室 |
| ip | NO | IP |
| page | YES | 页 |
| rpp | YES | 每页多少条数据 |

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | Int | 状态1:成功,0:失败 |
| ret | String | 消息信息 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": {

"totalNum": 20,

"list": [

{

"id": 1,

"building": "102",

"block": 2,

"room": 1001,

"ip": "192.168.1.100",

"sip": "8009",

"model":"DMT-100",

"remark": "新品"

},

{

"id": 2,

"building": "102",

"block": 2,

"room": 1002,

"ip": "192.168.1.101",

"sip": "8009",

"model": "DMT-100",

"remark": "新品"

}

]

}

}

* + **失败**

{

"state": 0,

"ret": "请求错误"

}

### 室内机修改

* URL：PUT /api/indoor/
* 参数：Json

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| id | YES | 唯一ID |
| model | YES | 型号 |
| sip | YES | SIP帐号 |
| ip | YES | IP |
| remark | NO | 备注 |

* 示例：

{

"id": "3",

"ip": "192.168.1.100",

"sip":"8008",

"model": "DMT-100",

"remark": "新款"

}

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | Int | 状态1：成功，0：失败 |
| ret | String | 消息信息 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败**

{

"state": 0,

"ret": ""

}

### 室内机删除

* URL：DEL /api/indoor/
* 参数：Json

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| id | YES | 唯一ID |

* 示例：

{

"id": 1

}

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | Int | 状态1：成功，0：失败 |
| ret | String | 消息信息 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败**

{

"state": 0,

"ret": ""

}

## 门口机

### 门口机录入

* URL：POST /api /outdoor
* 参数：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| building | YES | 栋 |
| block | YES | 单元 |
| room | YES | 号 |
| ip | YES | IP |
| model | YES | 型号 |
| remark |  | 备注 |

* 示例：

{

"building": "3",

"block": "2",

"room": "1",

"ip": "192.168.1.100",

"model": "DVP100",

"remark": "新款"

}

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | Int | 状态编码1:成功,0:失败 |
| ret | String | 返回结果 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败：**

{

"state": 0,

"ret": ""

}

### 门口机查询

* URL：GET /api/outdoor/ building=101&block=1&room=1&ip=192.168.1.100&page=2&rpp=20
* 参数：String

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| building | NO | 栋 |
| block | NO | 单元 |
| room | NO | 室 |
| ip | NO | IP |
| page | YES | 页 |
| rpp | YES | 每页多少条数据 |

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | Int | 状态1：成功，0：失败 |
| ret | String | 消息信息 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": {

"totalNum": 20,

"list": [

{

"id": 1,

"building": "102",

"block": 2,

"room": 1,

"ip": "192.168.1.100",

"sip": "8009",

"model":"DVP-100",

"remark": "新品"

},

{

"id": 2,

"building": "102",

"block": 2,

"room": 2,

"ip": "192.168.1.101",

"sip": "8009",

"model": "DVP-100",

"remark": "新品"

}

]

}

}

* + **失败**

{

"state": 0,

"ret": ""

}

### 门口机修改

* URL：PUT /api/outdoor/
* 参数：Json

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| id | YES | 唯一ID |
| model | YES | 型号 |
| ip | YES | IP |
| remark | NO | 备注 |

* 示例：

{

"id":"8",

"ip": "192.168.1.100",

"model": "DVP100",

"remark": "新款"

}

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | Int | 状态1：成功，0：失败 |
| ret | String | 消息信息 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败**

{

"state": 0,

"ret": ""

}

### 门口机删除

* URL：DEL /api/outdoor/
* 参数：Json

|  |  |  |
| --- | --- | --- |
| 字段名 | 是否必须 | 说明 |
| id | YES | 唯一ID |

* 示例：

{

"id": 1

}

* 返回：JSON

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| state | Int | 状态1：成功，0：失败 |
| ret | String | 消息信息 |

* 示例：
  + **成功：**

{

"state": 1,

"ret": ""

}

* + **失败**

{

"state": 0,

"ret": ""

}