OpenAPI

消息类型：

1. 平台信息设置
2. 设备信息查询
3. 设备控制命令

设置平台信息包括设置平台的登录名称以及登录密码。设置成功后，用户只需要根据平台名称和密码就可以登录平台从而获取家居环境下设备信息并控制。因为此平台是面向家居的，在一定程度上安全性是可以保障的，因此家居环境下的任何人，只要知道平台登录名称和密码，就可以接入平台，不用繁琐的用户注册过程。而且这样的设计也有很强的适应性，无论用户端使用手机app或者是网页等可视化方法，还是音箱等语音识别方法以达到连接平台控制设备的实现，这样的设计都可以支持。手机app或者是网页可以通过填写平台名称和密码登录平台，而音箱等语音设备则可以通过直接语音说出平台名称从而唤醒平台连接。

1. 平台信息设置

设置平台名称和密码

|  |  |
| --- | --- |
| 项目 | 描述 |
| url | http://tos.apiprovider.com/set |
| HTTP Method | POST |
| request | {“tosinfo”:  { “tos\_name”:”tos”,  “tos\_pwd”:”123”  }  } |
| response | {  “ack”:”ok”,  “error”:””  } |

登录平台

|  |  |
| --- | --- |
| 项目 | 描述 |
| url | http://tos.apiprovider.com/login |
| HTTP Method | POST |
| request | {  “tos\_name”:”tos”,  “tos\_pwd”:”123”  } |
| response | {  “ack”:”ok”,  “error”:””  } |

1. 设备信息查询

获得已注册设备列表

|  |  |
| --- | --- |
| 项目 | 描述 |
| URL | <http://tos.apiprovider.com/devices>/registered |
| HTTP Method | GET |
| response | {“deviceList”: [device\_id,device\_id,……..]} |

获得在线设备列表

|  |  |
| --- | --- |
| 项目 | 描述 |
| URL | <http://tos.apiprovider.com/devices>/online |
| HTTP Method | GET |
| response | {“deviceList”: [device\_id,device\_id,……..]} |

获得下线设备列表

|  |  |
| --- | --- |
| 项目 | 描述 |
| URL | <http://tos.apiprovider.com/devices>/offline |
| HTTP Method | GET |
| response | {“deviceList”: [device\_id,device\_id,……..]} |

设备信息查询

|  |  |
| --- | --- |
| 项目 | 描述 |
| URL | http://tos.apiprovider.com/devices/info/{device\_id} |
| HTTP Method | GET |
| response | {“device\_id”:”0xAFF0” ,  “device\_type”:”AirContainer”,  “properties”:{  “status”:”0x10”,  “temp”:”20”,  “windSpeed”:”30”,  “mode”:”0x50”  ……  }  } |

获得设备指令集

|  |  |
| --- | --- |
| 项目 | 描述 |
| URL | http://tos.apiprovider.com/devices/InstructionSet /{device\_id} |
| HTTP Method | GET |
| response | {“device\_id”:”0xAFF0” ,  “device\_type”:”LIGHT”,  “instructionSet”:{  “ctrlCode”:{  “turn\_on”：“0x11”，  “turn\_off”：“0x10”，  “light\_up”：“0x20”,  “light\_down”：“0x21”，  “change\_color”：“0x30”  },  “statusCode”:{  “on”:”0x11”,  “off”:”0x10”,  “light”:[0x20,0x21,0x22],  “color”:[0x30,0x31,0x32,0x33]  }  }  } |

获得设备联动策略列表

|  |  |
| --- | --- |
| 项目 | 描述 |
| URL | http://tos.apiprovider.com/linkCtrl/strategyList |
| HTTP  Method | GET |
| response | {  “strategyList”:[{  “strategy\_id”:xxx,  “strategyStatus”:”off”  },……]  } |

获得情景模式列表

|  |  |
| --- | --- |
| 项目 | 描述 |
| URL | http://tos.apiprovider.com/situationModeCtrl/modeList |
| HTTP  Method | GET |
| response | {  “modeList”:[{  “mode\_id”:xxx,  “modeStatus”:”off”  },……]  } |

1. 设备控制命令

单一设备控制

|  |  |
| --- | --- |
| 项目 | 描述 |
| URL | http://tos.apiprovider.com/ctrlDevice |
| HTTP  Method | POST |
| request | ｛  device\_id：xxx，  ctrlCode=xxx  ｝ |
| response | {  “ack”:”ok”,  “error”:””  } |

设备联动

设备联动策略注册

|  |  |
| --- | --- |
| 项目 | 描述 |
| URL | <http://tos.apiprovider.com/linkCtrl/register> |
| HTTP  Method | POST |
| request | {  “condition”:{  “conditionList”:[{  “device\_id”:xxx ,//作为条件的设备ID  “operator”:”eq”,  “expect\_value”:”90”  },……],  “relation”:”or”// 多个触发条件可以是或的关系，也 可以是与的关系  },  “actionList”:[{  “device\_id”:xxx,  “ctrlCode”:xxx  },……]  } |
| Response | {  “strategy\_id”:xxx,  “strategyStatus”:”off”  } |

启动设备联动策略

|  |  |
| --- | --- |
| 项目 | 描述 |
| URL | http://tos.apiprovider.com/linkCtrl |
| HTTP  Method | POST  ｛  strategy\_id=xxx，  strategyCtrl=on  ｝ |
| response | {  “ack”:”ok”,  “error”:””  } |

情景模式

情景模式注册(情景模式节点是由设备联动策略和单独控制的组合)

|  |  |
| --- | --- |
| 项目 | 描述 |
| URL | <http://tos.apiprovider.com/situationModeCtrl/register> |
| HTTP  Method | POST |
| request | {  “strategyList”:[{  “strategy\_id”:xxx ,  “strategyCtrl”:”on”  },……],  “actionList”:[{  “device\_id”:xxx,  “ctrlCode”:xxx  },……]  } |
| Response | {  “mode\_id”:xxx,  “modeStatue”:”off”  } |

启动情景模式

|  |  |
| --- | --- |
| 项目 | 描述 |
| URL | http://tos.apiprovider.com/situationModeCtrl |
| HTTP  Method | POST |
| request | ｛  mode\_id：xxx，  modeCtrl：on  ｝ |
| response | {  “ack”:”ok”,  “error”:””  } |