**服务器和小程序间的消息定义**

1. **小程序首先需要获取到自身的openId，服务器端口号8010；**
2. **小程序绑定插座。**

{

“method”:”up\_msg”,

“open\_id”:”XXXXXX”,

“req\_id”:123456789,

“ts”:12345678

“attr”:

{

“cmd”:”bind”,

“dev\_uuid”:” 02001122334455” /\* 中控设备的uuid \*/

}

}

**回复消息**

{

“method”:”down\_msg”,

“dev\_uuid”:”02001122334455”, /\* 子设备的uuid \*/

“req\_id”:123456789,

“code”:0 /\* 0成功 1无该设备 2该设备已经被绑定 \*/

}

1. **请求插座参数**

**请求消息**

{

“method”:”up\_msg”,

“open\_id”:”XXXXXX”,

“req\_id”:123456789,

“ts”:12345678

“attr”:

{

“cmd”:”get\_param”

}

}

**回复消息**

{

“method”:”down\_msg”,

“dev\_uuid”:”02001122334455”, /\* 中控设备的dev\_uuid \*/

“req\_id”:123456789,

“code”:0,

“attr”:

{

“dev1”:

{

“dev\_uuid”:”02001122334455”,

“switch”:”on”

},

“dev2”:

{

“dev\_uuid”:”02001122334455”,

“switch”:”on”

},

“dev3”:

{

“dev\_uuid”:”02001122334455”,

“switch”:”on”

},

“dev4”:

{

“dev\_uuid”:”02001122334455”,

“switch”:”on”

},

}

}

1. **设置插座开关**

**请求消息**

{

“method”:”up\_msg”,

“open\_id”:”XXXXXX”,

“dev\_uuid”:”02001122334455”,

“req\_id”:123456789,

“attr”:

{

“cmd”:”set\_switch”,

“dev\_uuid”:”02001122334455”, /\* 为子设备的uuid \*/

“switch”:”on”

}

}

**回复消息**

{

“method”:”down\_msg”,

“dev\_uuid”:”02001122334455”, /\* 子设备的uuid \*/

“req\_id”:123456789,

“code”:0

}

1. **主动上报消息**

{

“method”:”report\_msg”,

“dev\_uuid”:”10001122334455”,

“attr”:

{

“dev1”:

{

“dev\_uuid”:”02001122334455”,

“switch”:”on”

},

“dev2”:

{

“dev\_uuid”:”02001122334455”,

“switch”:”on”

},

“dev3”:

{

“dev\_uuid”:”02001122334455”,

“switch”:”on”

},

“dev4”:

{

“dev\_uuid”:”02001122334455”,

“switch”:”on”

},

}

}

子设备状态变化时服务器上报该消息至小程序该消息，该消息不需要回应。