LifeSmart Super Bowl Cloud Platform Service API (v1.12)

Version	revision date	Revised by	modify the content
0.1	2016/07/29	chenxuanye	first draft
0.2	2017/02/20	xiaoye	Add GetCustomKeys interface
0.3	2017/08/31	xiaoye	Add SendRemotesKeys interface
0.4	2017/10/12	Jon Fan	Organizing Documentation
0.5	2018/01/15	Jon Fan	added SetRemoteName interface
0.6	2018/09/17	Jon Fan	added GetRemote interface
0.7	2018/11/26	Pretty Zhu	added GetACRemoteState interface
1.11	2019/01/25	Jon Fan	organizes and arranges documents, adds ext_loc attribute support
1.12	2019/05/17	John Fan	Add GetRemoteFeature interface

directory

1. Preface

LifeSmart Super Bowl (SPOT) is a high-value household product launched by LifeSmart. It learns

The infrared remote control in life includes the complicated remote control Super Bowl, allowing users to experience a convenient life.

This article introduces the external interface of LifeSmart Super Bowl (SPOT), which enables third-party applications to operate the LifeSmart Super Bowl conveniently and quickly. Users can input category, brand, idx, and keys to the Sendkeys interface. A variable sends the infrared code to the Super Bowl to realize the control of the home appliance. The category, brand, and idx here can be obtained through GetCategory, GetBrands, and GetRemoteIdxs respectively, and idx and keys represent the remote control index and remote control keys respectively. And users can also use the LifeSmart Super Bowl external interface to add multiple remotes to the Super Bowl and manage the remotes. For more API introductions, please refer to the text.

LifeSmart, smart life, dreamer.

The APIs listed in this document are only for Super Bowl SPOT (universal remote control) operations. Please refer to "LifeSmart Cloud Platform Service Interface" for HTTPS request data format specifications and data signature methods. Among them GetCategory,

GetBrands, GetRemoteIdxs, GetCodes, GetACCodes methods

It does not involve user operations, so the userid and usertoken in the signature use the default values userid="10001" and usertoken="10001".

2. LifeSmart Super Bowl (SPOT) OpenAPI Interaction Flowchart

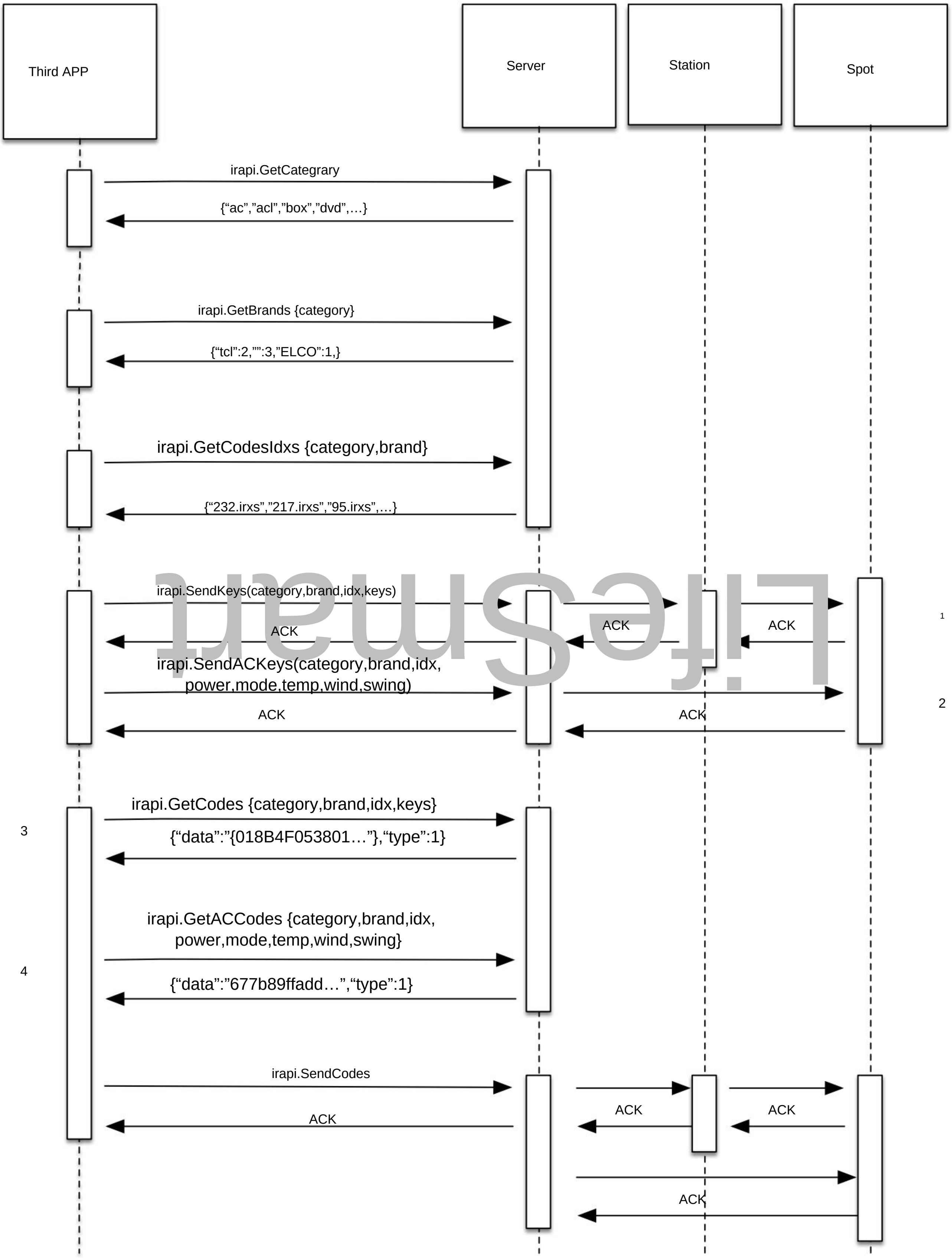
Third APP
Server
Station
Spot
irapi.GetCategrary
{“ac”,”acl”,”box”,”dvd”,…}
irapi.GetBrands {category}
{“tcl”:2,””:3,”ELCO”:1,}
irapi.GetCodesIdxs {category,brand}
{“232.irxs”,”217.irxs”,”95.irxs”,…}
irapi.SendKeys(category,brand,idx,keys)
ACK
ACK
ACK
1
irapi.SendACKeys(category,brand,idx,
power,mode,temp,wind,swing)
ACK
ACK
2
irapi.GetCodes {category,brand,idx,keys}
3
{“data”:”{018B4F053801…”},“type”:1}
irapi.GetACCodes {category,brand,idx,
power,mode,temp,wind,swing}
4
{“data”:”677b89ffadd…”,“type”:1}
irapi.SendCodes
ACK
ACK
ACK
ACK

Figure 1 Spot interaction process

Note 1 in the figure , Note 2 indicates that when the third-party application calls SendKeys (sends the infrared code of the remote control button) or SendACKeys (sends the infrared code of the air conditioner), the server (Server) can control the Super Bowl in two ways. Note 1 means that the server (Server) controls the Super Bowl through the LifeSmart Wisdom Center (Station); Note 2 means that the server can directly control the Super Bowl. Note 3 and Note 4 represent the APIs for obtaining ordinary infrared codes and air conditioner infrared codes respectively. Since the remote control of the air conditioner sends

Full code (that is, the infrared code contains data such as temperature and mode, not the infrared code of a certain button), so the API makes a distinction. In addition, it is recommended that third-party applications use SendKeys and SendACKeys. When in-depth customization is required, the GetCodes, GetACCodes, and SendCodes methods are used to control the Super Bowl.

3. API introduction of LifeSmart remote control

3.1.GetCategory Get the supported remote control types

3.1.1. JSON request data format

Type	Definition		Must	Description
Interface Name	GetCategory			Obtain the supported remote control types
Partial URL	irapi.GetCategory		Y	
Content Type	application/json		Y	
HTTP Method	HTTP POST		Y	
Request Content	system	to see	and	1.0
		just	And	in
		sign	Y	signature value
		userid	Y	This interface does not involve user operations, where the userid is a fixed value "10001", and the usertoken used for signing is also a fixed value "10001"
		appkey	Y	appkey
		did	O (optional)	the unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		Y	GetCategory
	id		Y	message id number

3.1.2. Examples

- We assume:

  appkey is APPKEY_XXXXXXXX, and real data needs to be filled in;
  apptoken is APPTOKEN_XXXXXXXX, and real data needs to be filled in; sign is SIGN_XXXXXXXX, and real signature data needs to be filled in;

- Request

  address: svrurl+PartialURL, for
  example: https://api.ilifesmart.com/app/irapi.GetCategory

- request info: {

```
        "id": 957,  "method":
        "GetCategory",  "system": {  "ver": "1.0",
        "lang": "en",  "userid": "10001",  "appkey":
             "APPKEY_XXXXXXXX",  "time":
             1447641115,  "sign": "SIGN_XXXXXXXX"







        }
   }
```

- Signature original string:

```
   method:GetCategory,time:1447641115,userid:10001,usertoken:10001,appk ey:APPKEY_XXXXXXXX,apptoken:APPTOKEN_XXXXXXXX
```

- Reply message:

```
   {
        "id": 957, "code":
        0, "message":
        ["ac","acl","box","stb","tv"
   }
```

Tip: type description

ac: air conditioner;

acl: air purifier; box:

network box; dvd: video

recorder; fan: fan; stb:

set-top box; tv: television

set;

3.2.GetBrands Get the corresponding type of remote control brand list

3.2.1. JSON request data format

Type	Definition		Must	Description
Interface Name GetBrands				Get the list of remote control brands of the corresponding type
Partial URL	irapi.GetBrands		Y	
Content Type	application/json Y			
HTTP Method	HTTP POST		Y	
Request Content	system	to see	Y	1.0
		just	Y	in
		sign	Y	signature value
		userid	Y	This interface does not involve user operations, where the userid is a fixed value "10001", and the usertoken used in signing is also a fixed value "10001"
		appkey	Y	appkey
		did	O	(Optional) The unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		Y	GetBrands
	params	category	Y	The category of the remote control, please check Description of the return message of GetCategory
	id		Y	message id number

3.2.2. Example •

We assume

that: appkey is APPKEY_XXXXXXXX, and actual data needs to be filled in; apptoken is APPTOKEN_XXXXXXXX, and actual data needs to be filled in; sign is SIGN_XXXXXXXX, and actual signature data needs to be filled in;

- Request

address: svrurl+PartialURL, for example: https://api.ilifesmart.com/app/irapi.GetBrands

- request info: {

```
  "id": 957,
  "method": "GetBrands",  "params":
  {  "category": "tv"  },  "system": {  "ver":
      "1.0",  "lang": "en",  "userid": "1111111",
  "appkey": "APPKEY_XXXXXXXX",  "time":
  1447641115,  "sign": "SIGN_XXXXXXXX"







  }
}
```

- Signature original string:

```
        method:GetBrands,category:tv,time:1447641115,userid:10001,usertoken: 10001,appkey:APPKEY_XXXXXXXX,apptoken:APPTOKEN_XXXXXXXX
```

- Reply message: {

```
            "id": 957, "code":
            0, "message": {

                "data": {"tcl":2,"Tag":3,"bosch":1,"Dongxinbao":4,"ELCO":2,…}, "params": {"category": "tv"}

            }
        }
```

Tip: The format of the returned data data is:
"data": {"brand": num,"brand":num,"brand":num, …}, "params": {…} num indicates that there are num remotes under the brand; params returns It is a user input parameter; for example: "tcl":2 indicates that there are 2 remote controls under the tcl brand, and "category": "tv" indicates that the parameter input by the user is to query the TV remote control.

3.3.GetRemoteIdxs Get the corresponding remote control index list

3.3.1. JSON request data format

Type	Definition		Must	Description
Interface Name GetRemoteIdxs				Get an indexed list of the corresponding remotes
Partial URL	irapi.GetRemoteIdxs Y			
Content Type	application/json		Y	
HTTP Method	HTTP POST		Y	
Request Content	system	to see	Y	1.0
		just	Y	in
		sign	Y	signature value
		userid	Y	This interface does not involve user operations, where the userid is a fixed value "10001", and the usertoken used in signing is also a fixed value "10001"
		appkey	Y	appkey
		did	O	(optional) the unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		Y	GetRemoteIdxs
	params	category Y		The category of the remote control, please check Return message of GetCategory
		brand		The brand of Y remote control, please check GetBrands return message
	id		Y	message id number

3.3.2. Examples

- We assume:

  appkey is APPKEY_XXXXXXXX, and real data needs to be filled in;
  apptoken is APPTOKEN_XXXXXXXX, and real data needs to be filled in; sign is SIGN_XXXXXXXX, and real signature data needs to be filled in;

- Request

  address: svrurl+PartialURL, for example:

```
https://api.ilifesmart.com/app/irapi.GetRemoteIdxs
```

- request info: {

```
    "id": 957,
    "method": "GetRemoteIdxs",  "params":
    {  "category":"tv", "brand":"tcl"


    },
    "system": {  "ver":
        "1.0",  "lang": "en",
        "userid": "10001",
        "appkey":
        "APPKEY_XXXXXXXX",  "time": 1447641115,
        "sign": "SIGN_XXXXXXXX"

    }
}
```

- Signature original string:

```
method:GetRemoteIdxs,brand:tcl,category:tv,time:1447641115,userid:10
001,usertoken:10001,appkey:APPKEY_XXXXXXXX,apptoken:APPTOKEN_XXXXXXXX
```

- Reply message:

```
{
    "code": 0, "id":
    957, "message":
    {
        "data":{"186.irxs","205.irxs","066.irxs", …}ÿ "params":
        {"category":"tv","brand":"tcl"}
    }
}
```

3.4.GetRemoteList Get the remote control list on the Super Bowl

3.4.1. JSON request data format

Type	Definition		Go t	Description
Interface Name	GetRemoteList			Get a list of remotes on the Super Bowl
Partial URL	irapi.GetRemoteList		Y	
Content Type	application/json		Y	
HTTP Method	HTTP POST		Y	
	system	to see	and	1.0
		just	And	in
		sign	Y	signature value
		userid	Y	User ID
		appkey	Y	appkey

Request Content		did	O	(optional) the unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		Y	GetRemoteList
	params	eight	Y	For the agt of the Super Bowl, see Return message of EpGetAllAgts
	id		Y	message id number

3.4.2. Examples

- We assume

  that: appkey is APPKEY_XXXXXXXX, and actual data needs to be filled in; apptoken is APPTOKEN_XXXXXXXX, and actual data needs to be filled in; usertoken is USERTOKEN_XXXXXXXX, and actual data needs to be filled in; sign is SIGN_XXXXXXXX, and actual signature data needs to be filled in;

- Request

  address: svrurl+PartialURL, for example: https://api.ilifesmart.com/app/irapi.GetRemoteList

- request info: {

```
    "id": 957,
    "method": "GetRemoteList",  "params": {  "agt":
    "_3MAAG1nYTAwMDA"  },  "system": {  "ver":
         "1.0",  "lang": "en",  "userid": "1111111",
    "appkey": "APPKEY_XXXXXXXX",  "time":
    1447641115,  "sign": "SIGN_XXXXXXXX"









    )
}
```

- Signature original string:

```
method:GetRemoteList,agt:_3MAAG1nYTAwMDA,time:1447641115,userid:1111
111,usertoken:USERTOKEN_XXXXXXXX,appkey:APPKEY_XXXXXXXX,apptoken:APP
TOKEN_XXXXXXXX
```

- Reply message:

```
{
     "code": 0, "id":
     957, "message":
     {
          "AI_IR_2718_1470020360":{"category":"box","brand":"apple",
               "name":"Apple Box RC","ts":1470020360},
          "AI_IR_2718_1470020405":{"category":"tv","brand":"tcl",
                "name":"TCL Remote","ts":1470020405},
            "AI_IR_2718_1470027934":
                {  "category":"tv",  "brand":"tcl",
                "name":"TCL Remote",
                "ts":1470027934,  "ext_loc":"{\"key\":
                \"LS\",\"location\":"HangZhou"}"

            }
        },
        "code":0,
        "id":816
    }
```

Tip: If there is an ext_loc attribute, the value of the ext_loc attribute will be returned.

3.5.AddRemote Add remote control to Super Bowl

3.5.1. JSON request data format

Type	Definition		Must	Description
Interface Name AddRemote				Add a remote to the Super Bowl
Partial URL	irapi.AddRemote		Y	
Content Type	application/json Y			
HTTP Method	HTTP POST		Y	
Request Content	system	to see	Y	1.0
		just	Y	in
		sign	Y	signature value
		userid	Y	User ID
		appkey	Y	appkey
		did	O	(Optional) The unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		Y	AddRemote
	params	eight	Y	For the super bowl agt in action, see Return message of EpGetAllAgts
		me	Y	To operate the Super Bowl me, please view Return message of EpGetAll
		category	Y	The category of the remote control to be added, please check the return message of GetCategory
		brand	Y	The brand of the remote control to be added, please check the return message of GetBrands
		idx	Y	For the idx of the remote to be added, please check Return message of GetRemoteIdxs
		name	Y	The name of the remote control to be added
		ext_loc	O	Remote control extension configuration to be set
	id		Y	message id number

3.5.2. Examples

- We assume that:

appkey is APPKEY_XXXXXXXX, and actual data needs to be filled in; apptoken is APPTOKEN_XXXXXXXX, and actual data needs to be filled in; usertoken is USERTOKEN_XXXXXXXX, and actual data needs to be filled in; sign is SIGN_XXXXXXXX, and actual signature data needs to be filled in;

- Request address:

  svrurl+PartialURL, for example: https:// api.ilifesmart.com/app/irapi.AddRemote

- request info: {

```
    "id": 957,  "method":
    "AddRemote",  "params": {  "agt":
    "_3MAAG1nYTAwMDA",  "me":"2718",
      "category": "tv",  "brand": "tcl",  "idx": "005.xs"
      "name": "TCLRemote" },

    "system": {  "ver":
          "1.0",  "lang": "en",
          "userid": "1111111",
          "appkey": "APPKEY_XXXXXXXX",
          "time": 1447641115,  "sign": "SIGN_XXXXXXXX"

    )
}
```

- Signature original string:

```
method:AddRemote,agt:_3MAAG1nYTAwMDA,brand:tcl,category:tv,idx:005.i
rxs,me:2718,name:TCLRemote,time:1447641115,userid:1111111,usertoken:
USERTOKEN_XXXXXXXX,appkey:APPKEY_XXXXXXXX,apptoken:APPTOKEN_XXXXXXXX
```

- Reply message:

```
{
    "code": 0, "id":
    957, "message":
    "AI_IR_2718_1470028017"
}
```

Note: If the command is executed successfully, that is, the code is equal to 0, the returned message is the ID of the created remote control.

3.6. DelRemote Delete Remote from Super Bowl

3.6.1. JSON request data format

Type	Definition		Go t	Description
Interface Name DelRemote				Remove the remote from the Super Bowl
Partial URL	irapi.DelRemote		Y	
Content Type	application/json		Y	
HTTP Method	HTTP POST		Y	
Request content	system	to see	and	1.0
		just	And	in
		sign	Y	signature value
		userid	Y	UserID
		appkey	Y	appkey
		did	O	(optional) the unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		And	DelRemote
	params	eight	Y	For the super bowl agt in action, see Return message of EpGetAllAgts
		id	Y	The id of the remote control to be deleted, please check Return message of GetRemoteList
	id		Y	message id number

3.6.2. Examples

- We assume

  that: appkey is APPKEY_XXXXXXXX, and actual data needs to be filled
  in; apptoken is APPTOKEN_XXXXXXXX, and actual data needs to be filled in;
  usertoken is USERTOKEN_XXXXXXXX, and actual data needs to be filled in;
  sign is SIGN_XXXXXXXX, and actual signature data needs to be filled in;

- Request

  address: svrurl+PartialURL, for
  example: https://api.ilifesmart.com/app/irapi.DelRemote

- request info: {

```
    "id": 957,
    "method": "DelRemote",  "params": {  "agt":
    "_3MAAG1nYTAwMDA",  "id":
       "AI_IR_2718_1470028017"  },  "system": {  "ver":
       "1.0",  "lang": "en",  "userid": "1111111",  "appkey":
    "APPKEY_XXXXXXXX",  "time": 1447641115,  "sign":
    "SIGN_XXXXXXXX"
    }
 }
```

- Signature original string:

```
method:DelRemote,agt:_3MAAG1nYTAwMDA,id:AI_IR_2718_1470028017,time:1
447641115,userid:1111111,usertoken:USERTOKEN_XXXXXXXX,appkey:APPKEY_
XXXXXXXX,apptoken:APPTOKEN_XXXXXXXX
```

- Reply message:

```
{
   "code": 0, "id":
   957, "message":"ok"
}
```

3.7.SetRemoteName Modify the name of the remote control on the Super Bowl

3.7.1. JSON request data format

Type	Definition		Go t	Description
Interface Name	SetRemoteName			Changing the name of the remote at the Super Bowl
Partial URL	irapi.SetRemoteName		Y	
Content Type	application/json		Y	
HTTP Method	HTTP POST		Y	
Request content	system	to see	and 1.0	
		just	And in	
		sign	Y	signature value
		userid	Y	UserID
		appkey	Y	appkey
		did	O	(optional) the unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		Y	SetRemoteName
	params	eight	Y	For the super bowl agt in action, see Return message of EpGetAllAgts
		eat	Y	The id of the remote control to be set, please check Return message of GetRemoteList
		name	Y	The name of the remote control to be set
	id		Y	message id number

3.7.2. Examples

- We assume

  that: appkey is APPKEY_XXXXXXXX, and actual data needs to be filled
  in; apptoken is APPTOKEN_XXXXXXXX, and actual data needs to be filled in;
  usertoken is USERTOKEN_XXXXXXXX, and actual data needs to be filled in; sign
  is SIGN_XXXXXXXX, and actual signature data needs to be filled in;

- Request

  address: svrurl+PartialURL, for
  example: https://api.ilifesmart.com/app/irapi.SetRemoteName

- request info: {

```
        "id": 957,  "method":
        "SetRemoteName",  "params": {  "agt":
        "_3MAAG1nYTAwMDA",  "id": "AI_IR_2718_1470028017",
             "name": "REMOTE_CONTROLLER",  },  "system":
             {  "ver": "1.0",  "lang": "en",  "userid": "1111111",  "appkey":
              "APPKEY_XXXXXXXX",  "time": 1447641115,  "sign":
         "SIGN_XXXXXXXX"









        }
    }
```

- Signature original string:

```
method:SetRemoteName,agt:_3MAAG1nYTAwMDA,id:AI_IR_2718_1470028017,na
me=REMOTE_CONTROLLER,time:1447641115,userid:1111111,usertoken:USERTO
KEN_XXXXXXXX,appkey:APPKEY_XXXXXXXX,apptoken:APPTOKEN_XXXXXXXX
```

- Reply message: {

```

        "code": 0, "id": 957,
        "message":"success"


    }
```

Tip: How to modify the ext_loc property of the remote?

The SetRemoteName interface supports modifying the value of the ext_loc attribute of the ˜remote controller. If you need to modify the ext_loc value of the remote control, please specify {agt,id,ext_loc} attributes; for example *{agt="* ext_loc="{\"key\": \"LS\", \"location\": "HangZhou"} "} indicates that the ext_loc *_3 EQQL 1 n YTA w MDA " , id = " AI _ AND _ 2718 _ 1470028017"* , attribute of the remote control should be modified to "{\"key\": \"LS\", \"location\": "HangZhou"}"; note: the ext_loc attribute can be modified together with the name attribute, Just specify their values at the same time.

3.8.SendKeys Send key commands of ordinary remote control

3.8.1. JSON request data format

Type	Definition		Must	Description
Interface Name SendKeys				Send key commands of ordinary remote control
Partial URL	irapi.SendKeys		Y	
Content Type	application/json		Y	
HTTP Method	HTTP POST		Y	
Request Content	system	to see	Y	1.0
		just	Y	in
		sign	Y	signature value
		userid	Y	User ID
		appkey	Y	appkey
		did	O	(Optional) The unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		Y	SendKeys
	params	eight	Y	For the super bowl agt in action, see Return message of EpGetAllAgts
		me	Y	To operate the Super Bowl me, please view Return message of EpGetAll
		category	Y	The category of the remote control to be operated, please check Return message of GetCategory
		brand	Y	The brand of the remote control to be operated, please check the return message of GetBrands
		idx	O	For the idx of the remote control to be operated, please check Return message of GetRemoteIdxs
		eat	O	The id of the remote control to be operated (the Super Bowl has been built), please check Return message of GetRemoteList
		keys	Y	The key value of the corresponding remote control needs to be converted into a list of JSON format strings, assigned to keys, and signed

	id	Y	message id number

ÿNotesÿ

- **keys: Multiple key values**

  can be filled in. For example: "keys": "[\"POWER\",\"1\",\"2\",\"3\"]" means that you will send to the Super Bowl Send "POWER", "1", "2", "3" four keys

- **idx and ai:** When

  the ai parameter exists, the key value is sent through the existing remote control. If there is no ai parameter, idx is sent to represent the general library to send the key value. One of the two must exist. Format such as:

```
"idx": "005.irxs" "ai":
"AI_IR_2713_1477029095"
```

3.8.2. Examples

- We assume

  that: appkey is APPKEY_XXXXXXXX, and actual data needs to be filled in; apptoken is APPTOKEN_XXXXXXXX, and actual data needs to be filled in; usertoken is USERTOKEN_XXXXXXXX, and actual data needs to be filled in; sign is SIGN_XXXXXXXX, and actual signature data needs to be filled in;

- Request

  address: svrurl+PartialURL for
  example: http://api.ilifesmart.com/app/irapi.SendKeys

- request info: {

```
    "id": 957,
    "method": "SendKeys",  "params":
    {  "agt": "_3MAAG1nYTAwMDA",
           "me": "2718",  "category": "tv", "brand":
           "tcl", "idx": "005.irxs", "keys": "[\"POWER\",
           \"1\",\"2\",\"3\"]"



    },
    "system": {  "ver":
         "1.0",  "lang": "en",
         "userid": "1111111",
         "appkey": "APPKEY_XXXXXXXX",
         "time": 1447641115,  "sign": "SIGN_XXXXXXXX"


    }
}
If the "ai" parameter exists, the params are as
  follows: "params": { "agt":
         "_3MAAG1nYTAwMDA", "me": "2718",
         "category": "tv",
           "brand": "tcl", "ai":
           "AI_IR_2713_1477029095", "keys": "[\"POWER\",
           \"1\",\"2\",\"3\"]"
 }
```

- Signature original

```
string: If idx method
is used: method:SendKeys,agt:_3MAAG1nYTAwMDA,brand:tcl,category:tv,idx:005.ir xs,keys:


["POWER","1","2" ,"3"],me:2718,time:1470023215,userid:1111111 ,usertoken:USERTOKEN_XXXXXXXX,appkey:APPKEY

ÿÿÿaiÿÿ:
method:SendKeys,agt:_3MAAG1nYTAwMDA,ai:AI_IR_2713_1477029095,brand:t cl,category:tv,keys:
["POWER","1","2","3"],me:2718,time:1470023215,us
erid:1111111,usertoken:USERTOKEN_XXXXXXXX,appkey:APPKEY_XXXXXXXX,app
token:APPTOKEN_XXXXXXXX
```

- Reply message:

```
{
    "code": 0, "id":
    957,
    "message":"ok"
}
```

3.9.SendACKeys sends air conditioner remote control commands

3.9.1. JSON request data format

Type	Definition		Must	Description
Interface Name	SendACKeys			Send air conditioner remote control command
Partial URL	irapi.SendACKeys		Y	
Content Type	application/json		Y	
HTTP Method	HTTP POST		Y	
	system	to see	Y	1.0
		just	Y	in
		sign	Y	signature value
		userid	Y	UserID
		appkey	Y	appkey
		did	O	(Optional) The unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		Y	SendACKeys
		eight	Y	agt for the super bowl, check out Return message of EpGetAllAgts
		me	Y	To operate the Super Bowl me, please view Return message of EpGetAll
		category Y		The category of the remote control to be operated, please check the return message of GetCategory
		brand	Y	The brand of the remote control to be operated, please check the return message of GetBrands
		idx	O	For the idx of the remote control to be operated, please check Return message of GetRemoteIdxs
		eat	O	The id of the remote control to be operated (the Super Bowl has been built), please check Return message of GetRemoteList

Request Content	params	key	Y	current key operation mode: power: power switch mode: mode switching temp: temperature change wind: wind speed change swing: wind direction change
		keyDetail O		Down temperature setting: "TEMP_DOWN", Up temperature setting: "TEMP_UP" Only control switch setting: "POWER_ONLY" The air conditioner is generally issued with a full code, and the infrared code will include the switch, mode, temperature, fan speed, etc. But there are some air conditioner code libraries, the switch and temperature, mode and other settings are separated. At this time, when the power is 1 and the key is "power", set the keyDetail to "POWER_ONLY" to turn on the air conditioner without setting the temperature , wind speed, etc. Note: This attribute distinguishes the codebase type, and unsupported codebases will ignore this parameter. **Note: Participate in signatures when present**
		power	Y	Switch(2) 1=on; 0=off
		mode	Y	Operation mode(5) 0=automatic 1=cooling 2=dehumidification 3=air supply 4=heating
		temp	Y	Temperature (15) 16-30 degrees
		wind	Y	Wind Speed(4) 0=Auto 1=Flow Speed 1 2=Flow Speed 2 3=Flow Speed 3
		swing	Y	Wind direction(5) 0=Auto 1=Wind direction 1 2=Wind direction 2 3=Wind direction 3 4=Wind direction 4
	id		Y	message id number

3.9.2. Examples

- We assume

  that: appkey is APPKEY_XXXXXXXX, and actual data needs to be

  filled in; apptoken is APPTOKEN_XXXXXXXX, and actual data needs to be

  filled in; usertoken is USERTOKEN_XXXXXXXX, and actual data needs to be

  filled in; sign is SIGN_XXXXXXXX, and actual signature data needs to be filled in;

- Request

  address: svrurl+PartialURL, for
  example: https://api.ilifesmart.com/app/irapi.SendACKeys

- request info: {

```
    "id": 957,
    "method": "SendACKeys",  "params":
    {  "agt": "_3MAAG1nYTAwMDA",
           "me": "2718",  "category": "ac", "brand":
           "aux", "idx": "33.irxs",  "key":"power",
           "power":1,  "mode":1,  "temp":25,
           "wind":3,  "swing":0

    },
    "system": {  "ver":
         "1.0",  "lang": "en",
         "userid": "1111111",
         "appkey": "APPKEY_XXXXXXXX",
         "time": 1447641115,  "sign": "SIGN_XXXXXXXX"

    }
}
```

If the "ai" parameter exists, the params are as
follows: "params": { "agt":

```
       "_3MAAG1nYTAwMDA", "me": "2718",
       "category": "ac", "brand": "aux", " ai":
       "AI_IR_2713_1477029095", "key":"power",
       "power":1, "mode":1, "temp":25, "wind":3,
       "swing":0

  }
```

- Signature original

```
string: If idx method
is used: method:SendACKeys,agt:_3MAAG1nYTAwMDA,brand:aux,category:ac,idx:33.i
rxs,key:power,me:2718,mode:1,power: 1,swing:0,temp:25,wind:3,time:147
0026010,userid:111111,usertoken:USERTOKEN_XXXXXXXX,appkey:APPKEY_XXX
XXXXX,apptoken:APPTOKEN_XXXXXXXX

ÿÿÿaiÿÿ:



method:SendACKeys,agt:_3MAAG1nYTAwMDA,ai:AI_IR_2713_1477029095,brand :aux,category:ac,key:power,me:2718,r
```

- Reply message:

```
{
    "code": 0, "id":
    957,
    "message":"ok"
}
```

3.10.GetCustomKeys Get a list of custom remote control keys

3.10.1. JSON request data format

Type	Definition		Must	Description
Interface Name	GetCustomKeys			Get a list of custom remote control buttons
Partial URL	irapi.GetCustomKeys		Y	
Content Type	application/json		Y	
HTTP Method	HTTP POST		Y	
Request Content	system	to see	Y	1.0
		just	Y	in
		sign	Y	signature value
		userid	Y	User ID
		appkey	Y	appkey
		did	O	(Optional) The unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		Y	GetCustomKeys
	params	eight	Y	agt for the super bowl, check out Return message of EpGetAllAgts
		eat	Y	The id of the custom remote that was created on Super Bowl, please check Return message of GetRemoteList
		category	Y	The category of the remote control, please check Return message of GetRemoteList
		brand	Y	The brand of remote control, please check Return message of GetRemoteList
	id		Y	message id number

3.10.2. Examples

- We assume

  that: appkey is APPKEY_XXXXXXXX, and actual data needs to be filled in; apptoken is APPTOKEN_XXXXXXXX, and actual data needs to be filled in; usertoken is USERTOKEN_XXXXXXXX, and actual data needs to be filled in; sign is SIGN_XXXXXXXX, and actual signature data needs to be filled in;

- Request

```
address: svrurl+PartialURL, for
example: https://api.ilifesmart.com/app/irapi.GetCustomKeys
```

- request info: {

```
    "id": 957,
    "method": "GetCustomKeys",  "params":
    {  "agt": "_3MAAG1nYTAwMDA",  "ai":
           "AI_IR_2713_1477029095"  "category":
           "custom",  "brand": "custom",


    },
    "system": {  "ver":
         "1.0",  "lang": "en",
         "userid": "1111111",
         "appkey": "APPKEY_XXXXXXXX",
         "time": 1447641115,  "sign": "SIGN_XXXXXXXX"


    )
}
```

- Signature original string:

```
method:GetCustomKeys,agt:_3MAAG1nYTAwMDA,ai:AI_IR_2713_1477029095,br
and:custom,category:custom,time:1470028470,userid:1111111,usertoken:
USERTOKEN_XXXXXXXX,appkey:APPKEY_XXXXXXXX,apptoken:APPTOKEN_XXXXXXXX
```

- Reply message:

```
{
    "code": 0, "id":
    957, "message":
    {

         "params":
              {  "agt":"_3MAAG1nYTAwMDA",
              "ai":"AI_IR_2713_1477029095",  "category":"custom"
              "brand":"custom"

         },
         "data":{"gh":"CS_3","g":"CS_2"}
    )
}
```

Tip: This interface is used to get the key list of the existing custom remote control. The calling process is as follows: 1. Call the GetRemoteList interface to get the remote control ai; 2. Call the GetCustomKeys interface to get the list of custom remote control keys , the data data of the return value is a key list, as in the above example: "gh" is the set key name, "CS_3" is the key key name; 3. Call the SendKeys interface to send the infrared code represented by the obtained key key name .

3.11.SendRemotesKeys Send multi-remote control and multi-key commands at one time

3.11.1. JSON request data format

Type	Definition		Go t	Description
Interface Name	SendRemotesKeys			Send multi-button commands of multiple remote controls to the Super Bowl at one time
Partial URL	irapi.SendRemotesKeys		Y	
Content Type	application/json		Y	
HTTP Method	HTTP POST		Y	request type
Request Content	system	to see	Y	1.0
		just	Y	in
		sign	Y	signature value
		userid	Y	User ID
		appkey	Y	appkey
		did	O	(optional) the unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		Y	SendRemotesKeys
	params	eight	Y	agt for the super bowl, check out Return message of EpGetAllAgts
		me	Y	Super Bowl me, check EpGetAll's return message
		keys	Y	Remote control information and key value information, multi-level tables do not participate in the signature
	id		Y	message id number

ÿNotesÿ

keys: Fill in the remote control information and key information to be sent, such as: "keys":

```
    [ {"category":"tv","brand":"changhong","ai":"AI_IR_2735_1503631196",
        "delay":1,"keys":["POWER","MENU"]},
    {"category":"ac","brand":"daikin","ai":"AI_IR_2735_1503631219",  "delay":1.2,"key":"power","power":1,"mode":0,
        "temp":27,"wind":2,"swing":0},


    {"category":"fan","brand":"ÿÿ","ai":"AI_IR_2735_1503631206",
        "delay":1,"keys":["POWER","WIND"]}
]
```

1. Each list item in keys must contain category (type of remote control), brand (brand of remote control), ai (ID of remote control); ,temp,wind,swing parameters (refer to SendACKeys); 3. Other remote control types must contain the keys parameter, keys is a list of key names, see the appendix; 4. The delay parameter is the interval between each key code to ensure the following They will not conflict with each other when sending, and the default delay time is 0.8s.

3.11.1. Examples

- We assume

  that: appkey is APPKEY_XXXXXXXX, and actual data needs to be filled in;

  apptoken is APPTOKEN_XXXXXXXX, and actual data needs to be filled in;

  usertoken is USERTOKEN_XXXXXXXX, and actual data needs to be filled in; sign is

  SIGN_XXXXXXXX, and actual signature data needs to be filled in;

- Request

  address: svrurl+PartialURL, for

  example: https://api.ilifesmart.com/app/irapi.SendRemotesKeys

- request info: {

```
    "id": 957,
    "method": "SendRemotesKeys",  "params":
    {  "agt": "_3MAAG1nYTAwMDA",  "me": "2735",
         "keys":{  {category="tv",brand="changhong",


                  ai="AI_IR_2735_1503631196",delay=1,
                  keys={"POWER","MENU"}},  {category="ac",brand="daikin",
              ai="AI_IR_2735_1503631219",delay=1.2,
                  key="power",power=1,mode=0,temp=27,wind=2,swing=0},
                {category="fan",brand="ÿÿ",  ai="AI_IR_2735_1503631206",delay=1,  keys={"POWER","WIND"}



         }
    },
    "system": {  "ver":
         "1.0",  "lang": "en",
         "userid": "1111111",
         "appkey": "APPKEY_XXXXXXXX",
         "time": 1447641115,  "sign": "SIGN_XXXXXXXX"


    }
}
```

- Signature original string: the default

secondary table does not participate in the signature

```
method:SendRemotesKeys,agt:_3MAAG1nYTAwMDA,me:2735,time:1447641115,u
serid:1111111,usertoken:USERTOKEN_XXXXXXXX,appkey:APPKEY_XXXXXXXX,ap ptoken:APPTOKEN_XXXXXXXX
```

- Reply message:

```
{
    "code": 0, "id":
    957, "message":
    "ok"
}
```

3.12.GetCodes Get the infrared codes of common remote control

3.12.1. JSON request data format

Type	Definition		Must	Description
Interface Name GetCodes				Obtain the infrared code of the common remote control
Partial URL	irapi.GetCodes		Y	
Content Type application/json Y				
HTTP Method	HTTP POST		Y	
Request Content	system	to see	Y	1.0
		just	Y	in
		sign	Y signature value	
		userid Y This interface does not involve user operations, the userid here is a		fixed value "10001", and the usertoken used in signing is also a fixed value "10001"
		appkey Y		appkey
		did	O	(Optional) The unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		Y	GetCodes
	params	category Y		For the category of the remote control to be inquired, please check Return message of GetCategory
		brand	Y The brand of the remote control to be inquired, please check	GetBrands return message
		idx	Y	For the idx of the remote controller to be queried, please check Return message of GetRemoteIdxs
		keys	O The key value of the corresponding remote control needs to be converted into a list	Assign the JSON format string to the keys and participate in the signature
	id		Y message id number	

ÿNotesÿ

- keys: The key value input format of

the **corresponding remote control key** is: "params": { "category": "tv",

```
        "brand": "tcl",  "idx":
        "005.irxs",  "keys": "[\"POWER\",
        \"1\",\"2\",\"3\"]"
    )
    If "key" is empty, that
    is, "params":
        { "category": "tv", "brand":
        "tcl", "idx": "005.irxs",



    } Then GetCodes will return the infrared codes of all keys, please refer to the appendix for the specific key table.
```

3.12.2. Examples

- We assume

  that: appkey is APPKEY_XXXXXXXX, and actual data needs to be filled

  in; apptoken is APPTOKEN_XXXXXXXX, and actual data needs to be filled in;

  usertoken is USERTOKEN_XXXXXXXX, and actual data needs to be filled in; sign is

  SIGN_XXXXXXXX, and actual signature data needs to be filled in;

- Request

  address: svrurl+PartialURL, for

  example: https://api.ilifesmart.com/app/irapi.GetCodes

- request info: {

```
        "id": 957,
        "method": "GetCodes",  "params":
        {  "category": "tv",  "brand": "tcl",
            "idx": "005.irxs",  "keys":
            "[\"POWER\",\"1\",\"2\",\"3\"]"  },
            "system": {  "ver": "1.0",  "lang":
            "en",  "userid": "10001",  "appkey": "APPKEY_XXXXXXXX",  "time":
        1447641115,  "sign": "SIGN_XXXXXXXX"








        )
    }
```

- Signature original string:

```
method:GetCodes,brand:tcl,category:tv,idx:005.irxs,keys:"[\"POWER\", \"1\",\"2\",
\"3\"]",time:1470026256,userid:10001,usertoken:10001,app key:APPKEY_XXXXXXXX,apptoken:APPTOKEN_XXXXXXXX
```

- Reply message:

```
    {
        "code": 0, "id":
        957,
            "message":  {


        "params":{"category":"tv","brand":"tcl","idx":"005.irxs","key":"4"}
               "codes": {  "2":
                   {"data":"018B4F0538016F4F3E57FF57FDFFFF5554
                               FF0001AD8B0360014F6F0340C2","type":1},
                   "3":{"data":"018B4F0538016F4F3E57FF57FF7FFDD554
                               FF0001AD8B0360014F6F0340C2","type":1},
                   "POWER":{"data":"018B4F0538016F4F3E57FF57FFDDFD7754
                               FF0001AD8B0360014F6F0340C2","type":1},
                   "1":{"data":"018B4F0538016F4F3E57FF57FFFFFD5554
                               FF0001AD8B0360014F6F0340C2","type":1}
               }
           }
        }
```

3.13.GetACCodes Get the infrared codes of the remote control of the air conditioner

3.13.1. JSON request data format

Type	Definition		Must	Description
Interface Name	GetACCodes			Obtain the infrared code of the remote control of the air conditioner
Partial URL	irapi.GetACCodes		Y	
Content Type	application/json		Y	
HTTP Method	HTTP POST		Y	
	system	to see	Y	1.0
		just	Y	in
		sign	Y	signature value
		userid	Y	This interface does not involve user operations, where the userid is a fixed value "10001", and the usertoken used for signing is also a fixed value "10001"
		appkey	Y	appkey
		did	O	(Optional) The unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		Y	GetACCodes
		category	Y	The category of the remote control, please check Return message of GetCategory
		brand	Y	The brand of the remote control, please check the return message of GetBrands

Request Content	params	idx	idx of Y remote control, please check	Return message of GetRemoteIdxs
		key	Y current key operation mode: power:	power switch mode: mode switching temp: temperature change wind: wind speed change swing: wind direction change
		power	Y	Switch(2) 1=on; 0=off
		mode	Y	Operation mode(5) 0=automatic 1=cooling 2=dehumidification 3=air supply 4=heating
		temp	Y	Temperature (15) 16-30 degrees
		wind	Y	Wind Speed(4) 0=Auto 1=Flow Speed 1 2=Flow Speed 2 3=Flow Speed 3
		swing	Y	Wind direction(5) 0=Auto 1=Wind direction 1 2=Wind direction 2 3=Wind direction 3 4=Wind direction 4
	id		Y	message id number

3.13.2. Examples

- We assume that:

appkey is APPKEY_XXXXXXXX, and actual data needs to be filled in; apptoken is APPTOKEN_XXXXXXXX, and actual data needs to be filled in; usertoken is USERTOKEN_XXXXXXXX, and actual data needs to be filled in; sign is SIGN_XXXXXXXX, and actual signature data needs to be filled in;

- Request address:

svrurl+PartialURL, for example:

```
https://api.ilifesmart.com/app/irapi.GetACCodes
```

- request info: {

```
        "id": 957,  "method":
        "GetACCodes",  "params": {  category":
        "ac",  "brand": "aux",  "idx": "33.irxs",
           "key":"power",  "power":1,  "mode":1,
           "temp":25,  "wind":3,  "swing":0,  },
           "system": {  "ver": "1.0",  "lang": "en",
           "userid": "10001",  "appkey":
           "APPKEY_XXXXXXXX",  "time":
           1447641115,  "sign": "SIGN_XXXXXXXX"

        }
    }
```

- Signature original string:

```
    method:GetACCodes,brand:aux,category:ac,idx:33.irxs,key:power,mode:1 ,power:1,swing:0,temp:25,wind:3,time:1447641115,userid:10001,usertok
     en:10001,appkey:APPKEY_XXXXXXXX,apptoken:APPTOKEN_XXXXXXXX
```

- Reply message:

```
    {
        "code": 0, "id":
        957, "message":
        [ {

              "params":{"category": "ac","brand": "aux","idx":  "33.irxs","key":"power",

                 "power":1, "mode":1, "temp": 25,"wind":3,"swing":0},  "codes":
              [  {"data":"018B4F0538016F4F3E57FF57FD7FFFD554

                       FF0001AD8B0360014F6F0340C2", "type":1}
              ]
           }
        ]
    }
```

3.14.SendCodes Super Bowl send infrared codes

3.14.1. JSON request data format

Type	Definition		Must	Description
Interface Name SendCodes				Super Bowl Launches Infrared Codes
Partial URL	irapi.SendCodes		Y	
Content Type	application/json		Y	
HTTP Method	HTTP POST		Y	
Request Content	system	to see	Y	1.0
		just	Y	in
		sign	Y	signature value
		userid	Y	User ID
		appkey	Y	appkey
		did	O	(Optional) The unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		Y	SendCodes
	params	eight	Y	To operate the agt of the super bowl, please check the return message of EpGetAllAgts
		me	Y	To operate the Super Bowl me, please view Return message of EpGetAll
		keys	Y	Infrared code, the list needs to be converted into a JSON format string and assigned to the keys to participate in the signature
	id		Y	message id number

Tips: keys: fill in one or more infrared codes to be sent, such as:

```
"keys":"[ {\"param\":{\"data\":\"018B4F0538016F4F3E57FF57FF7FFDD554
FF0001AD8B0360014F6F0340C2\",\"type\":1}}, {\"param\":
{\"data\":\"018B4F0538016F4F3E57FF57FF7FFDD554
FF0001AD8B0360014F6F0340C2\",\"type\":1}}, {\"param\":
{\"data\":\"018B4F0538016F4F3E57FF57FF7FFDD554
FF0001AD8B0360014F6F0340C2\",\"type\":1}} ]"
```

3.14.1. Examples

- We assume

  that: appkey is APPKEY_XXXXXXXX, and actual data needs to be filled
  in; apptoken is APPTOKEN_XXXXXXXX, and actual data needs to be filled in;
  usertoken is USERTOKEN_XXXXXXXX, and actual data needs to be filled in;
  sign is SIGN_XXXXXXXX, and actual signature data needs to be filled in;

- Request

  address: svrurl+PartialURL, for
  example: https://api.ilifesmart.com/app/irapi.SendCodes

- request info: {

```
        "id": 957,
        "method": "SendCodes",  "params":
        {  agt": "_3MAAG1nYTAwMDA",
             "me": "2718",  "keys":


     "[{\"param\":{\"data\":\"018B4F0538016F4F3E57FF57FF7FFDD554
     FF0001AD8B0360014F6F0340C2\",\"type\":1}}]"
        },
        "system": {  "ver":
             "1.0",  "lang": "en",
             "userid": "1111111",
             "appkey": "APPKEY_XXXXXXXX",
             "time": 1447641115, "sign": "SIGN_XXXXXXXX"


        }
     }
```

- Signature original string:

```
method:SendCodes,agt:_3MAAG1nYTAwMDA,keys:[{"param":{"data":"018B4F05380
16F4F3E57FF57FF7FFDD554FF0001AD8B0360014F6F0340C2","type":1}}],me:2718,t
ime:1447641115,userid:1111111,usertoken:USERTOKEN_XXXXXXXX,appkey:APPKEY
_XXXXXXXX,apptoken:APPTOKEN_XXXXXXXX
```

- Reply message:

```
     {
        "code": 0, "id":
        957, "message":
        "ok"
     }
```

3.15.GetRemote Get the remote control information on the Super Bowl

3.15.1. JSON request data format

Type	Definition	Must	Description
Interface Name GetRemote			Get remote info on the Super Bowl

Partial URL	irapi.GetRemote		Y	
Content Type	application/json		Y	
HTTP Method	HTTP POST		Y	
Request Content	system	to see	Y	1.0
		just	Y	in
		sign	Y	signature value
		userid	Y	User ID
		appkey	Y	appkey
		did	O	(Optional) The unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		Y	GetRemote
	params	eight	Y	To operate the agt of the super bowl, please check the return message of EpGetAllAgts
		eat	Y	The id of the remote control to be queried, please check Return message of GetRemoteList
		needKeys	Y	Whether it is necessary to return the key value of the remote control, the parameters have the following values: 0: no need to return the key value of the remote control; 1: return the array of remote control key names; 2: return the array of remote control key names, and also return the infrared corresponding to the remote control keys code and name (there is a name for the custom remote control);
	id		Y	message id number

3.15.1. Examples

- We assume

  that: appkey is APPKEY_XXXXXXXX, and actual data needs to be filled in; apptoken is APPTOKEN_XXXXXXXX, and actual data needs to be filled in; usertoken is USERTOKEN_XXXXXXXX, and actual data needs to be filled in; sign is SIGN_XXXXXXXX, and actual signature data needs to be filled in;

- Request

  address: svrurl+PartialURL, for example: https://api.ilifesmart.com/app/irapi.GetRemote

- request info: {

```
        "id": 957,  "method":
        "GetRemote",  "params": {  "agt":
        "A3EAAABtAEwQRzXXXXXXXX",  "ai":
            "AI_IR_2d14_1537154656",  "needKeys": 2


      },
        "system": {  "ver":
            "1.0",  "lang": "en",
            "userid": "1111111",
            "appkey": "APPKEY_XXXXXXXX",
            "time": 1447641115,  "sign": "SIGN_XXXXXXXX"


        }
    }
```

- Signature original string:

```
method:GetRemote,agt:A3EAAABtAEwQRzXXXXXXXX,ai:AI_IR_2d14_1537154656,nee
dKeys:2,time:1447641115,userid:1111111,usertoken:USERTOKEN_XXXXXXXX,appk ey:APPKEY_XXXXXXXX,apptoken:APPTOKEN_XXXXXXXX
```

- Reply message: {

```
        "code": 0, "id": 957,
        "message":
        { "name": "TV remote
            control", "category": "tv", "brand": "LG",
            "ts": 1537175081, " keys": ["POWER",
            "MUTE"], "codes": {


                "POWER": {  "type":
                    1,  "duty": 3,  "data":

                    "7F5F01556C03605854016_IR_DATA",  "name": "NAME"

                },
                "MUTE": {  "type":
                    1,  "duty": 3,  "data":

                    "7F5F01556C03605854016_IR_DATA",
                }
            },
        },
    }
```

Tip: If there is an ext_loc attribute, the value of the ext_loc attribute will be returned.

3.16.GetACRemoteState Get the current setting of the air conditioner remote control

3.16.1. JSON request data format

Type	Definition		Go t	Description
Interface Name	GetACRemoteState			Get the current settings of the air conditioner remote control
Partial URL	irapi.GetACRemoteState		Y	
Content Type	application/json		Y	
HTTP Method	HTTP POST		Y	
Request Content	system	to see	Y	1.0
		just	Y	in
		sign	Y	signature value
		userid	Y	User ID
		appkey	Y	appkey
		did	O	(optional) the unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		Y	GetACRemoteState
	params	eight	Y	To operate the agt of the super bowl, please check the return message of EpGetAllAgts
		eat	Y	The id of the remote control to be queried, please check Return message of GetRemoteList
	id		Y	message id number

3.16.1. Examples

- We assume

  that: appkey is APPKEY_XXXXXXXX, and actual data needs to be filled in; apptoken is APPTOKEN_XXXXXXXX, and actual data needs to be filled in; usertoken is USERTOKEN_XXXXXXXX, and actual data needs to be filled in; sign is SIGN_XXXXXXXX, and actual signature data needs to be filled in;

- Request

  address: svrurl+PartialURL, for

```
example: https://api.ilifesmart.com/app/irapi.GetACRemoteState
```

- request info: {

```
    "id": 957,
        "method": "GetACRemoteState",  "params":
        {  "agt": "A3EAAABtAEwQRzXXXXXXXX",
             "ai": "AI_IR_2d14_1537154656"

      },
        "system": {  "ver":
             "1.0",  "lang": "en",
             "userid": "1111111",
             "appkey": "APPKEY_XXXXXXXX",
             "time": 1447641115,  "sign":
             "SIGN_XXXXXXXX"

        }
    }
```

- Signature original string:

```
method:GetACRemoteState,agt:A3EAAABtAEwQRzXXXXXXXX,ai:AI_IR_2d14_1537154
656,time:1447641115,userid:1111111,usertoken:USERTOKEN_XXXXXXXX,appkey:A
PPKEY_XXXXXXXX,apptoken:APPTOKEN_XXXXXXXX
```

- Reply message:

```
    {
        "code": 0, "id":
        957, "message":
        {  "wind": 1,
             "swing": 0,
             "temp": 25,
             "mode": 1,  "key":
             "temp",  "power":
             0,  "keyDetail":
             "TEMP_DOWN"

        }
    }
```

3.17.GetRemoteFeature Get remote control features

3.17.1. JSON request data format

Type	Definition	Go t	Description
Interface Name	GetRemoteFeature		Obtain remote control features For air conditioner remote controls, you can obtain the supported capabilities, such as whether heating is supported, temperature setting range, etc. For non-air-conditioning remote controls, you can get their key-value collections, such as TV remote controls, whose key values are "POWER"...etc.
Partial URL	irapi.GetRemoteFeature Y		

Content Type application/json			Y	
HTTP Method	HTTP POST		Y	
Request Content	system	to see	Y	1.0
		just	Y	in
		sign	Y	signature value
		userid	Y	User ID
		appkey	Y	appkey
		did	O	(optional) the unique id of the terminal
		time	Y	UTC timestamp, the time calculated since January 1, 1970, in seconds
	method		Y	GetRemoteFeature
	params	category	Y	Remote control category
		brand	Y	Brand category, if querying the created remote control, this parameter can be filled with an empty string ""ÿ
		idx	O	The idx of the remote control, please provide this parameter if querying the remote control characteristics in the code library, otherwise provide the agt and ai parameters.
		sight		To query the ai of the Super Bowl, please provide this parameter if you want to query the remote control that has been created, otherwise just provide idx.
		eat	O	For the id of the remote control to be queried, please check the return message of GetRemoteList. If you want to query the remote control that has been created, please provide this parameter, otherwise provide idx.
	id		Y	message id number

3.17.1. Example of obtaining the characteristics of the remote control of the air conditioner

- We assume

  that: appkey is APPKEY_XXXXXXXX, and actual data needs to be filled in;

  apptoken is APPTOKEN_XXXXXXXX, and actual data needs to be filled in;

  usertoken is USERTOKEN_XXXXXXXX, and actual data needs to be filled in; sign is

  SIGN_XXXXXXXX, and actual signature data needs to be filled in;

- Request

  address: svrurl+PartialURL, for

  example: https://api.ilifesmart.com/app/irapi.GetRemoteFeature

- request info: {

```
      "id": 957,
      "method": "GetRemoteFeature",  "params":
      {  "category": "ac",  "brand": "DaiKin",  "idx": "10502"



      },
      "system": {  "ver":
           "1.0",  "lang": "en",
           "userid": "1111111",
           "appkey": "APPKEY_XXXXXXXX",
           "time": 1447641115,  "sign": "SIGN_XXXXXXXX"


      }
 }
```

- Signature original string:

```
method:GetRemoteFeature,brand:DaiKin,category:ac,idx:10502,time:14476411
15,userid:1111111,usertoken:USERTOKEN_XXXXXXXX,appkey:APPKEY_XXXXXXXX,ap ptoken:APPTOKEN_XXXXXXXX
```

- Reply message:

```
 {
      "code": 0, "id":
      957, "message":
      {  "code_mode":
           "full",  "frequency": 38000,  "ability":
           [ {


                     "temperature": "18,19,20,21,22,23,24,25,26,27,28,29,30",  "mode": 0,  "wind": "0,1,2,3"  },  {



                     "temperature": "18,19,20,21,22,23,24,25,26,27,28,29,30",  "mode": 1,  "wind": "0,1,2,3"  },  {




                     "mode": 2,
                     "wind": "0"
                },
                {
                     "temperature": "18,19,20,21,22,23,24,25,26,27,28,29,30",  "mode": 4,  "wind": "0,1,2,3"  },  {




                     "mode": 3,
                     "wind": "0,1,2,3"
                }

      ]  },}
```

• **Field description:**

ÿ The **code_mode** field indicates the infrared code encoding mode of the remote control of the air conditioner, and its values are: • **full** indicates that the infrared code is a full code, each code includes P_M_T_W (switch_mode_temperature_air volume), and a scene transmits a An infrared code is sufficient. • **power_independent**

indicates that the infrared code is not a full code, and POWER_ON/POWER_OFF is separated separately. For example, a combination code as follows: {'key':'power','power':1,'mode':1, 'temp':26,'wind':0} actually need to issue two infrared codes, the first is: POWER_ON, the second is M1_T26_W0. Of course, calling the irapi.SendACKeys method will automatically send two infrared codes, and the interval between the two infrared codes is 0.8 seconds. If the following combination code is issued: {'key':'mode','power':1,'mode':1, 'temp':26,'wind':0} and the air conditioner is already on, Then you only need to send an infrared code, because it adjusts the air conditioner mode and does not involve POWER, so only one infrared code is needed

M1_T26_W0ÿ

• **power_flip** flip is similar to power_independent, except that POWER_ON and POWER_OFF are the same infrared code, so calling POWER_ON twice will also turn off the air conditioner, and calling POWER_OFF twice will also turn on the air conditioner. Note: Even if it is a power_independent type, there is no guarantee that calling POWER_ON multiple times will not turn off the air conditioner in reverse.

• **keys** indicate that this type of air conditioner remote control is not a combination code type, that is, there is no full code such as P_M_T_W (switch_mode_temperature_flow), but each key represents a function, for example: POWER key means switch, press Press it once to turn it on, then press it again to turn it off; COOL key means to switch to cooling mode; TEMPDOWN/TEMPUP key means to lower/raise the temperature; this type of air conditioner remote control is similar to the TV type remote control, and cannot be directly set to a certain Mode, such as setting to "On_Cooling_26 degrees Celsius_Flow_Auto", must be set step by step through the remote control buttons. For the air conditioner remote control of the keys type, the returned data will contain the keys attribute, which lists the names of the supported keys. For example: {

```
"code": 0,
"message":
    {  "code_mode": "keys",  "keys": [

         "LR SWING",
         "TEMPDOWN",
         "TEMPUP",
         "At the Heat",
         "TIMER",
         "HEAT",
         "UD_WIND_MODE_SWING",
         "WIND_SPEED",
         "COOL",
         "DIGITAL STREAM",
         "POWER",
         "ENERGY SAVE",
         "DRY",
         "AIR PURIFYING"
    ],
    "frequency": 37910
}
}
```

ÿ The **frequency** field indicates the infrared frequency, generally 38000 ÿ The **ability** field indicates the capability of the remote control, and has the

following attributes: • **mode** indicates a currently supported mode, for example, 1 is cooling, 4 is heating, if a certain mode is not supported, then ability This mode will not be present in the array;

• **temperature** indicates the temperature selection supported in this mode, which is a string type, and each temperature is separated by a comma. If the set temperature is not supported, the value is empty, for example, there is no temperature setting in the dehumidification mode;

• **wind** indicates the wind volume selection supported in this mode, which is a string type, and each wind volume is separated by a comma. If the wind volume setting is not supported, the value is empty;

Note: If the code_mode is "keys", the air conditioner does not have the ability field.

ÿ The **keys** field indicates the key names supported by the air conditioner. Only air conditioners whose code_mode is "keys" have this field, for example: "keys": ["POWER", "TEMPDOWN", "TEMPUP", "COOL", "HEAT ", ...]

• **Tips: How to obtain the Feature of the air conditioner remote control that has been created?** If you want to get the Feature information of the created air conditioner remote control, you can provide the following parameters: {"category":"ac","brand":"DaiKin","agt":"A3EAAABtAEwQRzXXXXXXXX","ai:" AI_IR_2fbf_1558058988"}, the return value is the same as above.

• **Tips: How to use the air conditioner remote whose code_mode is "keys"?** The air conditioner remote whose code_mode is "keys" is used in a similar way to other non-air conditioner devices, such as TVs and set-top boxes. To transmit an infrared code, you need to call the irapi.SendKeys method, fill in "ac" for the category parameter, and fill in the corresponding keys for the keys parameter, for example: "keys": "[\"POWER\", \"COOL\"]", specify the sending Switch with refrigeration infrared code.

3.17.2. Example of getting TV remote control characteristics

• We assume that:

appkey is APPKEY_XXXXXXXX, and actual data needs to be filled in; apptoken is APPTOKEN_XXXXXXXX, and actual data needs to be filled in; usertoken is USERTOKEN_XXXXXXXX, and actual data needs to be filled in; sign is SIGN_XXXXXXXX, and actual signature data needs to be filled in;

• Request address:

svrurl+PartialURL, for example:

```
https://api.ilifesmart.com/app/irapi.GetRemoteFeature
```

• request info: {

```
"id": 957,
"method": "GetRemoteFeature",  "params":
{  "category": "tv",  "brand": "Sony",  "idx":
    "2327"


},
"system": {  "ver":
    "1.0",  "lang": "en",
    "userid": "1111111",
    "appkey": "APPKEY_XXXXXXXX",
    "time": 1447641115,  "sign":
    "SIGN_XXXXXXXX"
)
}
```

- Signature original string:

```
method:GetRemoteFeature,brand:Sony,category:tv,idx:2327,time:1447641115,
userid:1111111,usertoken:USERTOKEN_XXXXXXXX,appkey:APPKEY_XXXXXXXX,appto ken:APPTOKEN_XXXXXXXX
```

- Reply message:

```
     {
          "code": 0, "id":
          957, "message":
          {  "keys": [

                    "POWER",
                    "CYCLEWINDOWS",
                    "VOLUMEDOWN",
                    "VOLUMEUP",
                    "CHANNELDOWN",
                    "CHANNELUP",
                    ...
               ],
               "frequency": 38000  },

     }
```

- **Field description:**

ÿ The **keys** field indicates the key names currently supported by the remote control, please refer to the appendix

for details; ÿ The **frequency** field indicates the infrared frequency, generally 38000

- **Tips:** If

you want to get the Feature information of the created TV remote control, you can provide the following parameters: {"category":"tv","brand":"Sony","agt":"A3EAAABtAEwQRzXXXXXXXX","ai :"AI _IR_2fbf_1558058966"}, the return value is the same as above.

- **Tips:** Some remote

controllers press the same key twice in succession to flip the code. The code sent for the first time is different from the second time. The number is switched back and forth, and this code is defined as a flip code. For example, number 1: a&b (followed by the flip code) and number 2: c&d (followed by the flip code) Sending the number 11 means sending the code ab Sending the number 121 means sending the code ada Sending the number 12 means sending the code ad Sending the number 212 Just send the code cbc, send the number 111, and send the code ab a

Note 1: It is calculated according to the odd and even number of keys of this set of codes, & flipped back and forth. Note 2: Since we added this kind of code, we don't know whether the remote control sent the &before or the &before. So at the beginning of adding this set of codes, it is stipulated to start flipping from before &.

For remote controllers with flip codes, the keys list we return will list the flip codes, for example, key 1. If there is a flip code, the name is 1__sw, that is, the key name of the flip code is the original code + "__sw". Therefore, if you want to send 1,2,3, you need to call irapi.SendKeys("keys":"[\"1\", \"2__sw\", \"3\"]").

- For a remote control with a flip code, it is necessary to record the number of times the current button is issued on the caller's App side, and choose the original code or the flip code each time according to the parity of the number. For example, in the above example, if the current number of keystrokes is 11, you need to call irapi.SendKeys("keys":"[\"1__sw\", \"2\", \"3__sw\"] to send 1,2,3 ").

- For a remote control without a specific flip code, even if a key name with a flip code is issued, such as 2__sw, if the key value cannot be found in the code library, the original code 2 will still be used as the infrared of 2 by default. code value.

- **Prompt: How to judge whether there is a flip code?** Use

Ge to judge whether the keys collection returned by the tRemoteFeature interface contains the key name ending with "__sw". If so, it means that the key has a rollover code.

Appendix 1: TV remote control button list

TV key value (keys): TV	
keys	**paraphrase**
POWER	power supply
I can't	Li Yin
PARMUSIC	Accompanying sound
MODE	format
SLEEP	to sleep
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
0	0
-	—/-
SWAP	program alternation
EXCHANGE	exchange
PIP	picture in picture
NORMAL	normal
SELECT	channel selection
PICTURE	image
CHANNELUP	CH+
CHANNELDOWN	CH
AUDIO	voice
UP	superior
DOWN	Down
LEFT	left
RIGHT	right

TV key value (keys): TV	
MENU	menu
SCREEN	screen display
CYCLEWINDOWS	TV/Video
OK	Finish
CHANNELUP	channel+
CHANNELDOWN	channel
VOLUMEUP	volume+
VOLUMEDOWN	volume
MUTE	Mute

Appendix 2: Air Purifier Remote Control Button List

ACL button key value (keys): air purifier	
keys	**paraphrase**
POWER	power supply
AUTO	automatically
WIND	wind volume
TIMER	timing
ENERGY	energy saving
MUTE	Mute
FRESH	fresh
TASTE	deodorate
UP	superior
DOWN	Down
TIME	time
SAVE	keep

Appendix 3: DVD Remote Control Button List

DVD key value (keys): video recorder	
keys	**paraphrase**
POWER	power supply

DVD key value (keys): video recorder

PUSH	In/Out
MODE	format
SCREEN	screen display
VOLUMEUP	volume+
VOLUMEDOWN	volume
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10+	10+
0	0
RETURN	return
AUDIO	sound track
STOP	stop
PLAY	play
PAUSE	pause
PREV	previous song
NEXT	next song
FORWARD	fast forward
BACKWARD	rewind
SETUP	set up
UP	superior
DOWN	Down
LEFT	left
RIGHT	right
MENU	menu
OK	Sure

DVD key value (keys): video recorder	
LANGUAGE	language
PROGRAM	programming
MUTE	Mute
TITLE	title
SUBTITLE	subtitle

Appendix 4: Set-top box remote control button list

STB button key value (keys): set-top box	
1	1
2	2
3	3
4	4
5	5
	7
	7
8	8
9	9
0	0
LIST	the list
_UN1	previous show
_UN2	previous show
POWER	power supply
CHANNELUP	channel+
CHANNELDOWN	channel
VOLUMEUP	volume+
VOLUMEDOWN	volume
UP	superior
DOWN	Down
LEFT	left
RIGHT	right

STB button key value (keys): set-top box	
OK	confirm
EXIT	quit
MENU	menu
RED	red
GREEN	green
YELLOW	ÿ
BLUE	blue
RETURN	return
PAGEUP	previous page
PAGEDOWN	next page
AUDIOCHN	sound track
INFO	information
MUTE	Mute
FAV	favorite
GUIDE	guide
TV	television
FM	broadcast
INFOR	Information
STOCK	stock
DEMAND	on demand
EMAIL	mail
GAME	game
LIST2	the list
SETUP	set up
HOME	home page
RECORD	record •
STOP	stop
A	A
B	B
C	C
D	D
AND	AND

STB button key value (keys): set-top box	
F	F
FORWARD	fast forward
BACKWARD	rewind
PLAY	play / Pause
_REV1	reserve
_REV2	reserve
_REV3	reserve
_REV4	reserve
_REV5	reserve
_REV6	reserve

Appendix 5: Network box remote control button list

Box button key value (keys):	
key	paraphrase
POWER	ON
UP	superior
DOWN	Down
LEFT	left
RIGHT	right
OK	confirm
MENU	menu
HOME	home page
RETURN	return
VOLUMEUP	VOL+
VOLUMEDOWN	VOL
SETUP	set up

Appendix 6: Button List of Fan Remote Control

FAN button key value (keys): fan

keys	paraphrase
POWER	ON
MODE	wind mode
SWAY	shaking head warm air
WIND	On/wind speed
SOIL	light/cooling/humidification
HEATH	Health wind
ANION	Negative ion/power saving
TIMER	Timing/Appointment