

tuya
Device Control
Cloud Services
API Reference



tuya Device Control Cloud Services API Reference User Guide

[Home](#) » [Tuya](#) » tuya Device Control Cloud Services API Reference User Guide 

Contents

- 1 [tuya Device Control Cloud Services API Reference](#)
- 2 [Product Information](#)
- 3 [Product Usage Instructions](#)
- 4 [Get the instruction set by category](#)
- 5 [Get the instruction set by device](#)
- 6 [Get the instruction set by devices](#)
- 7 [Specifications and properties of the device](#)
- 8 [Send instructions to the device](#)
- 9 [Get the latest device status](#)
- 10 [Documents / Resources](#)
 - 10.1 [References](#)
- 11 [Related Posts](#)



tuya Device Control Cloud Services API Reference



Product Information

The Device Control product provides a set of APIs and instructions for controlling devices remotely. Users can send instructions to the device using the provided API endpoints.

Specifications:

- Device Control
- **Version:** 20240124
- Online Version

Product Usage Instructions

1. Get the Instruction Set by Category:

Users can retrieve the instruction set by category using the following API endpoints:

- API Description
- API Address
- Request Parameter
- Return Parameter
- Request Example
- SDK Example
- Response Example
- Error Code

2. Get the Instruction Set by Device:

To obtain the instruction set by the device, users can utilize the provided API endpoints for specific devices. This includes details such as API address, request parameters, return parameters, and examples.

3. Get the Instruction Set by Devices:

This feature allows users to access the instruction set for multiple devices. Users can make requests using the

API address and parameters provided.

4. Get the Specifications and Properties of the Device:

This API endpoint provides detailed specifications and properties of the device, including the instruction set and status set. Users can request this information along with examples of how to use the SDK.

5. Send Instructions to the Device:

Users can send specific instructions to the device using the provided API. This includes details such as request parameters, return parameters, and examples of how to send instructions effectively.

FAQ:

- **Q: How can I troubleshoot if I encounter errors when sending instructions?**

A: If you encounter errors when sending instructions, refer to the error code section in the API documentation to identify and troubleshoot the issue.

- **Q: Can I control multiple devices simultaneously?**

A: Yes, you can control multiple devices simultaneously by utilizing the instructions set by the device feature and making requests for each device accordingly.

This topic describes the APIs related to device control, including getting instruction sets, specifications, properties, the latest statuses, and sending instructions.

Request method	API	Description
GET	/v1.0/functions/{category}	Get the instruction set by category
GET	/v1.0/devices/{device_id}/functions	Get the instruction set by device
GET	/v1.0/devices/{device_id}/: Get the device' s specifications and properties (including instruction set and status set)	
POST	/v1.0/devices/{device_id}/commands	Send instructions to the device
GET	/v1.0/devices/{device_id}/: Get the latest device status	

Get the instruction set by category

API description

Query the instruction set by category. This instruction set has the biggest amount of instructions in the category of all-in-one apps, which can be used and referred to by developers. If you are a platform developer, you can develop applications according to this instruction set.

API address

1. GET /v1.0/ functions /{ category}

Request parameter

Parameter name	Data type	Parameter type	Required	Description
category	String	URI	Yes	Category name, such as kg, cz, and dj.

Return parameter

Parameter name	Data type	Description
code	Integer	Error code .
success	Boolean	Determine whether the request succeeded or not. true: succeeded false: failed
Parameter name	Data type	Description
msg	String	The message returned if the request fails. It will return null if the request succeeds.
result	Object< result >	The returned result.

• Description of result

- | Parameter name | Category | Description |
- | :—| :—| :—|
- | category| String | Product category |
- | functions| List<funnctions> | Instruction set |

• Description of functions

- | Parameter name | Category | Description |
- | :—| :—| :—|
- | code | String | Instruction | | type | String | Instruction type |
- | values | String | The range of input parameter values |
- | name | String | Instruction name |
- | desc | String | Description |

Request example

1. GET /v1.0/ functions/kg

SDK example

```
1 TuyaClient client = new TuyaClient(clientId, secret, RegionEnum.CN);
2 CategoryFunctions categoryFunctions = client.getFunctionByCategory("
3 kg");
4 System.out.println("Get the function list by category: ");
5 System.out.println(JSONObject.toJSONString(categoryFunctions  ));
```

Response example

```
1 {
2     "success": true,
3     "t": 1571293457435,
4     "result": {
5         "category": "kg",
6         "functions": [
7             {
8                 "name": "Switch",
9                 "desc": "The master switch. true: Open all sub switch
10 hes. false: Close all sub switches",
11                 "code": "switch",
12                 "type": "Boolean",
13                 "values": "{}"
14             },
15             {
16                 "name": "Switch 1",
17                 "desc": "Switch 1",
18                 "code": "switch_1",
19                 "type": "Boolean",
20                 "values": "{}"
21             },
22             {
23                 "name": "Switch 2",
24                 "desc": "Switch 2",
25                 "code": "switch_2",
26                 "type": "Boolean",
27                 "values": "{}"
28             },
29             {
30                 "name": "Switch 3",
31                 "desc": "Switch 3",
32                 "code": "switch_3",
33                 "type": "Boolean",
34                 "values": "{}"
35             },
36             {
37                 "name": "Switch 4",
38                 "desc": "Switch 4",
39                 "code": "switch_4",
40                 "type": "Boolean",
41                 "values": "{}"
42             },
43             {
44                 "name": "Switch 5",
45                 "desc": "Switch 5",
46                 "code": "switch_5",
47                 "type": "Boolean",
48                 "values": "{}"
49             },
50             {
51                 "name": "Switch 6",
```

```

1      "desc": "Switch 6",
2      "code": "switch_6",
3      "type": "Boolean",
4      "values": "{}"
5  },
6  {
7      "name": "The countdown of switch 1",
8      "desc": "The countdown of switch 1",
9      "code": "countdown_1",
10     "type": "Integer",
11     "values": "{\unit\":"s","\min\:0,\max\:86400,\
12 "scale\:0,\step\:1}"
13 },
14 {
15     "name": "The countdown of switch 2",
16     "desc": "The countdown of switch 2",
17     "code": "countdown_2",
18     "type": "Integer",
19     "values": "{\unit\":"s","\min\:0,\max\:86400,\
20 "scale\:0,\step\:1}"
21 },
22 {
23     "name": "The countdown of switch 3",
24     "desc": "The countdown of switch 3",
25     "code": "countdown_3",
26     "type": "Integer",
27     "values": "{\unit\":"s","\min\:0,\max\:86400,\
28 "scale\:0,\step\:1}"
29 },
30 {
31     "name": "The countdown of switch 4",
32     "desc": "The countdown of switch 4",
33     "code": "countdown_4",
34     "type": "Integer",
35     "values": "{\unit\":"s","\min\:0,\max\:86400,\
36 "scale\:0,\step\:1}"
37 },
38 {
39     "name": "The countdown of switch 5",
40     "desc": "The countdown of switch 5",
41     "code": "countdown_5",
42     "type": "Integer",
43     "values": "{\unit\":"s","\min\:0,\max\:86400,\
44 "scale\:0,\step\:1}"
45 },
46 {
47     "name": "The countdown of switch 6",
48     "desc": "The countdown of switch 6",
49     "code": "countdown_6",
50     "type": "Integer",
51     "values": "{\unit\":"s","\min\:0,\max\:86400,\

```

```

1  "scale\":0,\step\:1}"
2      },
3      {
4          "name": "The countdown of USB 1",
5          "desc": "The countdown of USB 1",
6          "code": "countdown_usb1",
7          "type": "Integer",
8          "values": "{\unit\":"s\","min\:0,\"max\:86400,\"
9  "scale\":0,\step\:1}"
10     },
11     {
12         "name": "The countdown of USB 2",
13         "desc": "The countdown of USB 2",
14         "code": "countdown_usb2",
15         "type": "Integer",
16         "values": "{\unit\":"s\","min\:0,\"max\:86400,\"
17 "scale\":0,\step\:1}"
18     },
19     {
20         "name": "USB1 ",
21         "desc": "USB 1",
22         "code": "switch_usb1",
23         "type": "Boolean",
24         "values": "{}"
25     },
26     {
27         "name": "USB 2",
28         "desc": "USB 2",
29         "code": "switch_usb2",
30         "type": "Boolean",
31         "values": "{}"
32     }
33 ]
34 }
35 }

```

Error code

The following are common service exceptions for this interface. For more exception errors, see [Error Code](#).

Error code	Description
500	System error
2015	This category is not supported at present

Get the instruction set by device

API description

Query the functions supported by the device. You can send control by the obtained instructions.

API address

1. GET /v1.0/ devices /{ device_id }/ functions

Request parameter

Parameter name	Data type	Parameter type	Required	Description
device_id	String	URI	Yes	Device ID

Return parameter

Parameter name	Data type	Description
code	Integer	Error code .
success	Boolean	Determine whether the request succeeded or not. true: succeeded false: failed
msg	String	The message returned if the request fails. It will return null if the request succeeds.

Parameter name	Data type	Description
result	Object< result >	The returned result.

• Description of result

- | Parameter name | Category | Description |
- | :—| :—| :—|
- | category| String | Product category |
- | functions| List<function> | Instruction set |

• Description of functions

- | Parameter name | Category | Description |
- | :—| :—| :—|
- | code | String | Instruction |
- | type | String | Instruction type |
- | values | String | The range of input parameter values |
- | name | String | Instruction name |
- | desc | String | Description |

Request example

1. GET /v1.0/ devices/vdevo155XXX83954683/functions

SDK example

1. TuyaClient client = new TuyaClient(clientId , secret , RegionEnum.CN);
2. categoryFunctions = client.getFunctionsByDevId(DEV_ID);
3. System.out.println("Get the function list by device ID: ");
4. System.out.println(JSONObject.toJSONString(category Functions));

Response example

```
1  {
2      "success": true,
3      "t": 1571293776551,
4      "result": {
5          "category": "cz",
6          "functions": [
7              {
8                  "name": "Switch",
9                  "desc": "Switch",
10                 "code": "switch_1",
11                 "type": "Boolean",
12                 "values": "{}"
13             },
14             {
15                 "name": "The countdown of switch 1",
16                 "desc": "The countdown of switch 1",
17                 "code": "countdown_1",
18                 "type": "Integer",
19                 "values": "{\"unit\":\"s\",\"min\":0,\"max\":86400,\"scale\"
20 :0,\"step\":1}"
21             }
22         ]
23     }
24 }
```

Error code

The following are common service exceptions for this interface. For more exception errors, see [Error Code](#).

Error code	description
500	System error
1106	Illegal permission

Get the instruction set by devices

API description

Query the functions supported by the devices List. You can send control by the obtained instructions. support up to 20 devices

API address

1. GET /v1.0/ devices/functions

Request parameter

Parameter name	Data type	Parameter type	Required	Description
device_ids	String	URI	Yes	Devices List

Return parameter

Parameter name	Data type	Description
code	Integer	Error code .
success	Boolean	Determine whether the request succeeded or not. true: succeeded false: failed
msg	String	The message returned if the request fails. It will return null if the request succeeds.
Parameter name	Data type	Description
result	Object< result >	The returned result.

• Description of result

- | Parameter name | Category | Description |
- | :—| :—| :—|
- | category| String | Product category |
- | functions| List<function> | Instruction set |
- | devices | List | device Id List |

• Description of functions

- | Parameter name | Category | Description |
- | :—| :—| :—|
- | code | String | Instruction |
- | type | String | Instruction type |
- | values | String | The range of input parameter values |

- | name | String | Instruction name |
- | desc | String | Description |

Request example

1. GET /v1.0/ devices/functions?device_ids =06200286 dc4f22c8418c ,6c8b eb2f
2. a517fbc901dtln

SDK example

1. TuyaClient client = new TuyaClient(clientId , secret , RegionEnum.CN);
2. categoryFunctions = client.getFunctionsByDevsList(DEV_ID);
3. System.out.println("Get the function list by devices List: ");
4. System.out.println(JSONObject.toJSONString(category Functions));

Response example

```

1  {
2      "result": [
3          {
4              "category": "cz",
5              "devices": [
6                  "06200286dc4f22c8418c"
7              ],
8              "functions": [
9                  {
10                     "code": "switch",
11                     "desc": "开关",
12                     "name": "开关",
13                     "type": "Boolean",
14                     "values": "{}"
15                 },
16                 {
17                     "code": "countdown_1",
18                     "desc": "开关1倒计时",
19                     "name": "开关1倒计时",
20                     "type": "Integer",
21                     "values": "{\"unit\":\"s\",\"min\":0,\"max\":86400,\"scale
22 \":0,\"step\":1}"
23                 }
24             ]
25         }
26     ],
27     "success": true,
28     "t": 1618822009199
29 }

```

Error code

The following are common service exceptions for this interface. For more exception errors, see [Error Code](#).

Error code	description
500	System error
1106	Illegal permission

Specifications and properties of the device

Get the specifications and properties of the device, including the instruction set and status set

API description

This interface is used to get the instruction set and status set supported by the device according to the device ID.

API address

1. GET /v1.0/ devices /{ device_id }/ specifications

Request parameter

Parameter name	Data type	Parameter type	Required	Description
device_id	String	URI	Yes	Device ID

Return parameter

Parameter name	Data type	Description
code	Integer	Error Code .
success	Boolean	Determine whether the request succeeded or not. true: succeeded false: failed
msg	String	The message returned if the request fails. It will return null if the request succeeds.
Parameter name	Data type	Description
result	Object< result >	The returned result.

- **Description of result**

- | Parameter name | Category | Description |
- | :—| :—| :—|
- | category | String | Product category |
- | functions | List<functions> | Instruction set |
- | status | List<status> | Status set |

- **Description of functions**

- | Parameter name | Category | Description |
- | :—| :—| :—|
- | code | String | command |
- | type | String | Type |
- | values | String | The range of parameter values |

- **Description of status**

- | Parameter name | Category | Description |
- | :—| :—| :—|
- | code | String | status code |
- | type | String | Type |
- | values | String | The range of parameter values |

Request example

- GET /v1.0/ devices/vdevoXXXXXXXXXXXXXXXXX/specifications

SDK example

N/A

Return parameter

```

1 {
2   "success": true,
3   "t": 1571201730542,
4   "result": {
5     "category": "dj",
6     "functions": [
7       {
8         "code": "switch_led",
9         "type": "Boolean",
10        "values": "{}"
11      },
12      {
13        "code": "work_mode",
14        "type": "Enum",
15        "values": "{\"range\": [\"colour\"]}"
16      },
17      {
18        "code": "bright_value",
19        "type": "Integer",
20        "values": "{\"min\":0,\"scale\":0,\"unit\":\"\", \"ma
21 x\":255,\"step\":1}"
22      },
23      {
24        "code": "colour_data",
25        "type": "Json",
26        "values": "{\"h\":{\"min\":1,\"scale\":0,\"unit\":\"
27 \",\"max\":360,\"step\":1},\"s\":{\"min\":1,\"scale\":0,\"un  it\":\"
28 \",\"max\":255,\"step\":1},\"v\":{\"min\":1,\"scale\":0,\"uni  t\":\"
29 \",\"max\":255,\"step\":1}}"
30      }
31    ],
32    "status": [
33      {
34        "code": "switch_led",
35        "type": "Boolean",
36        "values": "{}"
37      },
38      {
39        "code": "work_mode",
40        "type": "Enum",
41        "values": "{\"range\": [\"colour\"]}"
42      },
43      {
44        "code": "bright_value",
45        "type": "Integer",
46        "values": "{\"min\":0,\"scale\":0,\"unit\":\"\", \"ma
47 x\":255,\"step\":1}"
48      },
49      {
50        "code": "colour_data",
51        "type": "Json",
52        "values": "{\"h\":{\"min\":1,\"scale\":0,\"unit\":\"
53 \",\"max\":360,\"step\":1},\"s\":{\"min\":1,\"scale\":0,\"un  it\":\"
54 \",\"max\":255,\"step\":1},\"v\":{\"min\":1,\"scale\":0,\"uni  t\":\"
55 \",\"max\":255,\"step\":1}}"
56      }
57    ]
58  }
59 }

```

Send instructions to the device

API description

This interface is used to send one or more obtained instruction sets. Whether instruction sets can be sent simultaneously depends on the product.

API address

1. POST /v1.0/ devices /{ device_id }/ commands

Request parameter

Parameter name	Data type	Parameter type	Required	Description
device_id	String	URI	Yes	Device ID
commands	Object	body	Yes	Command set

Return parameter

Parameter name	Data type	Description
code	Integer	Error code .
success	Boolean	Determine whether the request succeeded or not. true: succeeded false: failed
msg	String	The message returned if the request fails. It will return null if the request succeeds.

Parameter name	Data type	Description
result	Boolean	Determine whether the request succeeded or not.

Request example

1. POST /v1.0/ devices/vdevo1552XXXX3954683/commands

```

1  {
2      "commands":[
3          {
4              "code":"switch_led",
5              "value":true
6          },
7          {
8              "code":"bright",
9              "value":30
10         }
11     ]
12 }

```

SDK example

```

1  TuyaClient client = new TuyaClient(clientId, secret, RegionEnum.CN);
2  JSONObject temp = new JSONObject();
3  temp.put("v", 0);
4  temp.put("s", 1);
5  temp.put("h", 1);
6  Command command = new Command("colour_data", temp);
7  List<Command> list = new ArrayList<Command>(1);
8  list.add(command);
9  Boolean isSuccess = client.postDeviceCommand("vdevo155013689282523",
10 list);
11 System.out.println("Send device control instructions: ");
12 System.out.println(isSuccess);

```

Response example

```

1  {
2      "success":true,
3      "t":1551851043862,
4      "result":true
5  }

```

Error code

The following are common service exceptions for this interface. For more exception errors, see [Error Code](#).

Error code	Description
500	System error
2008	Instruction not supported. Confirm whether the instruction or value is correct

Get the latest device status

API description

Get the latest device status by the device ID

API address

1. GET /v1.0/ devices /{ device_id }/ status

Request parameter

Parameter name	Data type	Parameter type	Required	Description
device_id	String	URI	Yes	Device ID

Return parameter

Parameter name	Data type	Description
code	Integer	Error code .
success	Boolean	Determine whether the request succeeded or not. true: succeeded false: failed
msg	String	The message returned if the request fails. It will return null if the request succeeds.
Parameter name	Data type	Description
result	Boolean	Determine whether the request succeeded or not.

Request example

1. GET /v1.0/ devices /{ device_id }/ status

SDK example

```

1 TuyaClient client = new TuyaClient(clientId, secret, RegionEnum.CN);
2 List<Status> deviceStatus= client.getDeviceStatus(DEV_ID);
3 System.out.println("Get device status: ");
4 System.out.println(JSONObject.toJSONString(deviceStatus));

```

Response example

```

1 {
2     "success": true,
3     "t": 1545447665981,
4     "result": [
5         {
6             "code": "switch_led",
7             "value": true
8         },
9         {
10            "code": "work_mode",
11            "value": "scene_2"
12        },
13        {
14            "code": "bright_value",
15            "value": 25
16        },
17        {
18            "code": "temp_value",
19            "value": 0
20        },
21        {
22            "code": "colour_data",
23            "value": "{\"h\":37.0,\"s\":255.0,\"v\":189.0}"
24        },
25        {
26            "code": "scene_data",
27            "value": ""
28        },
29        {
30            "code": "flash_scene_1",
31            "value": ""
32        },
33        {
34            "code": "flash_scene_2",
35            "value": ""
36        },
37        {
38            "code": "flash_scene_3",
39            "value": ""
40        },
41        {
42            "code": "flash_scene_4",
43            "value": "{\"bright\":255,\"frequency\":80,\"hsv\":{\"h\":0.0,\"s\":255.0,\"v\":255.0},{\"h\":120.0,\"s\":255.0,\"v\":255.0},{\"h\":240.0,\"s\":255.0,\"v\":255.0},{\"h\":300.0,\"s\":255.0,\"v\":255.0},{\"h\":240.0,\"s\":255.0,\"v\":255.0},{\"h\":0.0,\"s\":255.0,\"v\":255.0}],\"temperature\":255}"
44        }
45    ]
46 }

```

```

1     ]
2 }

```

Error code

The following are common service exceptions for this interface. For more exception errors, see [Error Code](#).

Error code	Description
500	System error

Documents / Resources

	tuya Device Control Cloud Services API Reference [pdf] User Guide Device Control Cloud Services API Reference, Cloud Services API Reference, API Reference
---	---

References

- [User Manual](#)

Manuals+. [Privacy Policy](#)

This website is an independent publication and is neither affiliated with nor endorsed by any of the trademark owners. The "Bluetooth®" word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. The "Wi-Fi®" word mark and logos are registered trademarks owned by the Wi-Fi Alliance. Any use of these marks on this website does not imply any affiliation with or endorsement.