ShadeConnector openAPI v1

Updated date: 2020/12/18

Summary

ShadeConnector *openAPI* (Hereinafter referred to as *openAPI*) is a Restful API, it provides devices/scenes discovery and device/scenes control abilities for the 3rd party to integrate window covering devices via Cloud to Cloud.

- The Connector APP is requested to add, edit, and delete devices/scenes.
- An extra bridge is required for bi-directional devices and one way devices, while Wi-Fi integrated devices do not.
- Before operating *openAPI*, your server IP/IPs must be added into the *openAPI* server whitelist and you will be assigned a unique identifier (`appKey` and `appSecret`) .

For more questions, please contact your sales consultant or email info@dooya.com.

openAPI list

The following table shows the supported interfaces.

HTTP method	*URL*	*Description*
POST	https://openapi.shadeconnector.com/v1/app/oauth/token	*Client* token creation
POST	https://openapi.shadeconnector.com/v1/app/oauth/deleteToken	*Client* token deletion
POST	https://openapi.shadeconnector.com/v1/app/oauth/refreshToken	Refresh *Client* token
POST	https://openapi.shadeconnector.com/v1/user/login	*User* login
POST	https://openapi.shadeconnector.com/v1/user/logout	*User* logout
POST	https://openapi.shadeconnector.com/v1/user/refreshToken	Refresh *User* token
POST	https://openapi.shadeconnector.com/v1/user/getAreasWithDevices	Get device list
POST	https://openapi.shadeconnector.com/v1/user/device/control	Device control
POST	https://openapi.shadeconnector.com/v1/user/device/info	Get device status query
POST	https://openapi.shadeconnector.com/v1/user/scenes	Get scene list
POST	https://openapi.shadeconnector.com/v1/user/scene/trigger	Scene control

Client token creation

`POST` `https://openapi.shadeconnector.com/v1/app/oauth/token` is used to verify the identity of the integrator (Hereinafter referred to as the *Client*). Each *Client* will be assigned a group of `appKey` and `appSecret`.

`appKey` and `appSecret` are used to crate the *client* `accessToken`, the *client* `accessToken` will be used as `H-APP-Token` in the following interfaces for *user* login/logout, devices/scenes discovery, and devices/scenes control.

Pay attention that **DO NOT** share your `appKey` and `appSecret` with anyone else.

Request

Parameter name	***Type***	***Required***	***Value***	***Description***
`appKey`	string	Yes		*Client* Key
`sign`	string	Yes		sign = HMAC-SHA256(`appKey` + `t` , `appSecret`).toUpperCase() Use `appKey` and `t` (10-digit timestamp of current time) to concatenate the string to be signed, and use `appSecret` as the key to participate in the hash digest, the resulting string is finally capitalized
`signMethod`	string	Yes	*HMAC-SHA256*	Please fill in *HMAC-SHA256*
`t`	long	Yes		10-digit timestamp (YY-MM-DD hh:mm:ss)(Accurate to the second)

Response

Parameter name	***Type***	***Description***
`code`	int	Status code
`msg`	string	Status
`data`	object	Data

`data`

Parameter name	***Type***	***Description***
`accessToken`	string	The *client* `accessToken` is used as `H-APP-Token` in the following interfaces.
`refreshToken`	string	The *client* `refreshToken` is used to refresh *client* `accessToken` . The validity period of *client* `refreshToken` is 4 hours.
`expiresIn`	int	Validity period of `accessToken` (in seconds)

Response example

```
{
    "msg": "OK",
    "code": 20000,
    "data": {
        "expiresIn": 7200,
        "accessToken": "1a2b3d4e5f1a2b3d4e5f1a2b3d4e5f1a",
        "refreshToken": "1a2b3d4e5f1a2b3d4e5f1a2b3d4e5f1a"
    }
}
```

Client token deletion

`POST` `https://openapi.shadeconnector.com/v1/app/oauth/deleteToken`

After deleting, all end *user*s can no longer operate devices/scenes through *openAPI* under this *client* `accessToken`, and this *client* `accessToken` and its *client* `refreshToken` will expire.

Request

Parameter name	***Type***	***Required***	***Value***	***Description***
`appKey`	string	Yes		*Client* Key
`sign`	string	Yes		sign = HMAC-SHA256(`appKey` + `t` , `appSecret`).toUpperCase() Use `appKey` and `t` (10-digit timestamp of current time) to concatenate the string to be signed, and use `appSecret` as the key to participate in the hash digest, the resulting string is finally capitalized.
`signMethod`	string	Yes	*HMAC-SHA256*	Please fill in *HMAC-SHA256*
`t`	long	Yes		10-digit timestamp (YY-MM-DD hh:mm:ss)(Accurate to the second)
`accessToken`	string	Yes		Please fill in the 'to be deleted' *client* `accessToken`

Response

Parameter name	***Type***	***Description***
`code`	int	Status code
`msg`	string	Status

Response example

```
{
    "msg": "OK",
    "code": 20000
}
```

Refresh *Client* token

`POST` `https://openapi.shadeconnector.com/v1/app/oauth/refreshToken` is used to refresh the *client* `accessToken` after expiration.

Pay attention the *client* `refreshToken` will also be refreshed when calling `POST` `https://openapi.shadeconnector.com/v1/app/oauth/refreshToken` . After refreshing, the previous *client* `refreshToken` will expire.

Request

Parameter name	***Type***	***Required***	***Value***	***Description***
`appKey`	string	Yes		*client* Key
`sign`	string	Yes		sign = HMAC-SHA256(`appKey` + `t` , `appSecret`).toUpperCase() Use `appKey` and `t` (10-digit timestamp of current time) to concatenate the string to be signed, and use `appSecret` as the key to participate in the hash digest, the resulting string is finally capitalized.
`signMethod`	string	Yes	*HMAC-SHA256*	Please fill in *HMAC-SHA256*
`t`	long	Yes		10-digit timestamp (YY-MM-DD hh:mm:ss)(Accurate to the second)
`refreshToken`	string	Yes		*client* `refreshToken`

Response

Parameter name	***Type***	***Description***
`code`	int	Status code
`msg`	string	Status
`data`	object	Data

`data`

Parameter name	***Type***	***Description***
`accessToken`	string	The *client* `accessToken` is used as `H-APP-Token` in the following interfaces.
`refreshToken`	string	The *client* `refreshToken` is used to refresh *client* `accessToken`. The validity period of *client* `refreshToken` is 14 days.
`expiresIn`	int	Validity period of `accessToken` (in seconds)

Response example

```
{
    "msg": "OK",
    "code": 20000,
    "data": {
        "expiresIn": 7200,
        "accessToken": "1a2b3d4e5f1a2b3d4e5f1a2b3d4e5f1a",
        "refreshToken": "1a2b3d4e5f1a2b3d4e5f1a2b3d4e5f1a"
    }
}
```

User login

`POST` `https://openapi.shadeconnector.com/v1/user/login`, the end *user* login interface. `username` (the user account/email address) and `password` are the same as those used to log in the Connector APP.

After *user* `accessToken` expires, you need to call `POST` `https://openapi.shadeconnector.com/v1/user/refreshToken` to refresh a new *user* `accessToken`.

Headers

Parameter name	***Type***	***Required***	***Description***
`H-APP-Token`	string	Yes	*client* `accessToken`

Body

Parameter name	***Type***	***Required***	***Description***
`username`	string	Yes	The 'Connector' APP account/email address
`password`	string	Yes	The 'Connector' App password

Notes
`password` is MD5 hash encryption and converted to a 32-byte uppercase string.

For instance, fill *E10ADC3949BA59ABBE56E057F20F883E* in `password` if the original password is *123456*.

Response

Parameter name	*Type*	*Description*
`code`	int	Status code
`msg`	string	Status
`data`	object	Data

`data`

Parameter name	*Type*	*Description*
`accessToken`	string	*user* `accessToken`
`refreshToken`	string	*user* `refreshToken`, the validity period of `refreshToken` is 14 days.
`expiresIn`	int	validity period of *user* `accessToken` (in seconds)

Response Example

```
{
    "msg": "OK",
    "code": 20000,
    "data": {
        "expiresIn": 604800,
        "accessToken": "1a2b3d4e5f1a2b3d4e5f1a2b3d4e5f1a",
        "refreshToken": "1a2b3d4e5f1a2b3d4e5f1a2b3d4e5f1a"
    }
}
```

User logout

`POST` `https://openapi.shadeconnector.com/v1/user/logout`, the *user* logout interface. After logging out, the *user* can no longer operate devices/scenes via *openAPI*.

Headers

Parameter name	*Type*	*Required*	*Description*
`H-APP-Token`	string	Yes	*client* `accessToken`

Body

Parameter name	*Type*	*Required*	*Description*
`username`	string	Yes	The 'Connector' APP account/email address
`accessToken`	string	Yes	*user* `accessToken`

Response

Parameter name	*Type*	*Description*
`code`	int	Status code
`msg`	string	Status

Response Example

```
{
    "code": 20000,
    "msg": "OK"
}
```

Refresh *User* token

`POST` `https://openapi.shadeconnector.com/v1/user/refreshToken` is used to refresh *user* `accessToken` after expiration.

Note that *user* `refreshToken` will also be refreshed when calling `POST` `https://openapi.shadeconnector.com/v1/user/refreshToken` . After refreshing, the previous *user* `refreshToken` will expire.

Headers

Parameter name	*Type*	*Required*	*Description*
`H-APP-Token`	string	Yes	*client* `accessToken`

Body

Parameter name	*Type*	*Required*	*Description*
`accessToken`	string	Yes	*user* `accessToken`
`refreshToken`	string	Yes	*user* `refreshToken`

Response

Parameter name	*Type*	*Description*
`code`	int	Status code
`msg`	string	Status
`data`	object	Data

`data`

Parameter name	*Type*	*Description*
`accessToken`	string	*user* `accessToken`
`refreshToken`	string	*user* `refreshToken`, the validity period of `refreshToken` is 14 days.
`expiresIn`	int	validity period of *user* `accessToken` (in seconds)

Response example

```
{
    "msg": "OK",
    "code": 20000,
    "data": {
        "expiresIn": 604800,
        "accessToken": "1a2b3d4e5f1a2b3d4e5f1a2b3d4e5f1a",
        "refreshToken": "1a2b3d4e5f1a2b3d4e5f1a2b3d4e5f1a"
    }
}
```

Get Device List

`POST` `https://openapi.shadeconnector.com/v1/user/getAreasWithDevices` is used to poll the list of devices under a specific user.

Headers

Parameter name	*Type*	*Required*	*Description*
`H-APP-Token`	string	Yes	*client* `accessToken`

Body

Parameter name	*Type*	*Required*	*Description*
`accessToken`	string	Yes	*user* `accessToken`

Response

Parameter name	*Type*	*Description*
`code`	int	Status code
`msg`	string	Status
`data`	object	Data

`data`

Parameter name	*Type*	*Description*
`areas`	array	`areas` is the 'Location' in the Connector APP

`areas`

Parameter name	*Type*	*Description*
`areaCode`	string	Location code
`areaName`	string	Location name
`rooms`	object	Room information
`devices`	object	Devices under the location

Notes:

Generally `devices` under `areas` array only includes bridges, unless child devices or Wi-Fi integrated devices have no room allocated.

`rooms`

Parameter name	*Type*	*Description*
`roomCode`	string	Room code
`roomName`	string	Room name
`devices`	object	Devices under rooms

`devices`

Parameter name	*Type*	*Description*
`mac`	string	Device ID
`deviceType`	string	deviceType
`deviceAlias`	string	Device alias (it's the same device name in the Connector APP)
`deviceData`	object	Devices configurations

`deviceData`

Parameter name	*Type*	*Value*	*Description*
`currentPosition`	int	0~100	Device current lift position
`currentAngle`	int	0~180	Device current tilt position
`operation`	int		0: Close/Down 1: Open/Up 2: Stop 5: Status query
`type`	int		type
`batteryLevel`	int		Power voltage (DC motor only)
`voltageMode`	int	0/1	0: AC Motor 1: DC Motor

Response example


```
{
    "msg": "OK",
    "code": 20000,
    "data": {
        "areas": [
            {
                "areaCode": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6f",
                "rooms": [
                    {
                        "devices": [],
                        "roomCode": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6f-1-
1a2b3c4d",
                        "roomName": "Bedroom"
                    },
                    {
                        "devices": [
                            {
                                "deviceType": "100",
                                "deviceAlias": "Left blind",
                                "deviceData": {
                                    "currentPosition": 70,
                                    "type": 2,
                                    "voltageMode": 1,
                                    "operation": 2,
                                    "currentAngle": 0,
                                    "batteryLevel": 824
                                },
                                "mac": "1a2b3c4d5e6f7g8h"
                            },
                            {
                                "deviceType": "100",
                                "deviceAlias": "One-way blind",
                                "deviceData": {
                                    "type": 1,
                                    "operation": 2
```

```
                                },
                                "mac": "1a2b3c4d5e6f7g8i"
                            }
                        ],
                        "roomCode": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6f-1-
1a2b3c4e",
                        "roomName": "Living room"
                    }
                ],
                "areaName": "MyHouse",
                "devices": [
                    {
                        "deviceType": "0",
                        "deviceAlias": "HomeBridge",
                        "deviceData": {},
                        "mac": "1a2b3c4d5e6f"
                    }
                ]
            }
        ]
    }
}
```


Device Control

Control a specific device.
`POST` `https://openapi.shadeconnector.com/v1/user/device/control`

Headers

Parameter name	*Type*	*Required*	*Description*
`H-APP-Token`	string	Yes	*client* `accessToken`

Body

Parameter name	*Type*	*Value*	*Required*	*Description*
`accessToken`	string		Yes	*user* `accessToken`
`mac`	string		Yes	Device ID
`deviceType`	string		Yes	Please fill in the correct `deviceType` to control the device
`targetPosition`	string	0~100	Optional	Device lift target position
`targetAngle`	string	0~180	Optional	Device tilt target position
`operation`	string	0|1|2|5	Optional	0: Close/Down 1: Open/Up 2: Stop 5: Status query

Notes

1. `deviceType` must match the device Type obtained from the Get Device List interface.
2. All devices support `operation` *Value* 0|1|2 .
3. One-way motors/receivers do not support `targetPosition` , `targetAngle` and `operation` *Value* 5
4. `operation` and `targetPosition`/`targetAngle` can not be used at the same time in the same request.
5. `targetPosition` and `targetAngle` can be used at the same time in the same request.
6. Not all devices support `targetAngle` , it's based on `deviceType` and `type` .
7. `operation` *Value* 5 is used to poll the device status from the device side.

Response

Parameter name	*Value*	*Description*
`code`	int	Status code
`msg`	string	Status

Response example

```
{
    "msg": "OK",
    "code": 20000
}
```

Device status query

`POST` `https://openapi.shadeconnector.com/v1/user/device/info` is used to query the device shadow status from the server side.

Headers

Parameter name	*Type*	*Required*	*Description*
`H-APP-Token`	string	Yes	*client* `accessToken`

Body

Parameter name	*Type*	*Required*	*Description*
`accessToken`	string	Yes	*user* `accessToken`
`mac`	string	Yes	Device ID
`deviceType`	string	Yes	Device type

Response

Parameter name	*Type*	*Description*
`code`	int	Status code
`msg`	string	Status
`data`	object	Data

`data`

Parameter name	*Type*	*Description*
`mac`	int	Device ID
`deviceType`	string	Device type
`deviceAlias`	string	Device name
`deviceData`	object	Data

`deviceData`

Parameter name	*Type*	*Value*	*Description*
`currentPosition`	int	0~100	Current lift position
`currentAngle`	int	0~180	Current tilt position
`operation`	int		0: Close/Down 1: Open/Up 2: Stop 5: Status query
`type`	int		1:Roller Blinds 2:Venetian Blinds 3:Roman Blinds 4:Honeycomb Blinds 5:Shangri-La Blinds 6:Roller Shutter 7:Roller Gate 8:Awning 10:Day&night Blinds 11:Dimming Blinds 12:Curtain 13:Curtain(right stick) 14:Curtain(left stick)
`batteryLevel`	int		Power voltage (DC motor only), actual voltage x 100.
`voltageMode`	int	0|1	0: AC Motor 1: DC Motor

Response Example

```
{
    "msg": "OK",
    "code": 20000,
    "data": {
        "deviceType": "100",
        "deviceAlias": "Left blind",
        "deviceData": {
            "currentPosition": 70,
            "type": 2,
            "voltageMode": 1,
            "operation": 2,
            "currentAngle": 0,
            "batteryLevel": 823
        },
        "mac": "1a2b3c4d5e6f7g8h"
    }
}
```

Get Scene List

`POST` `https://openapi.shadeconnector.com/v1/user/scenes` is used to get the list of scenes.

Headers

Parameter name	*Type*	*Required*	*Description*
`H-APP-Token`	string	Yes	*client* `accessToken`

Body

Parameter name	*Type*	*Required*	*Description*
`accessToken`	string	Yes	*user* `accessToken`

Response

Parameter name	*Type*	*Value*	*Description*
`code`	int	Status code	Status code
`msg`	string	Status	Status
`data`	object	Data	

`data`

Parameter name	*Type*	*Description*
`scenes`	array	Scene configures

`scenes`

Parameter name	*Type*	*Description*
`areaCode`	string	Location code
`sceneCode`	string	Scene code
`sceneName`	string	Scene name

Response example

```
{
    "msg": "OK",
    "code": 20000,
    "data": {
        "scenes": [
            {
                "areaCode": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6f",
                "sceneCode": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6f",
                "sceneName": "Good morning"
            },
            {
                "areaCode": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6g",
                "sceneCode": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6g",
                "sceneName": "Bed time"
            }
        ]
    }
}
```

Scene Control

`POST` `https://openapi.shadeconnector.com/v1/user/scene/trigger` is used to activate a specific scene.

Headers

Parameter name	*Type*	*Required*	*Description*
`H-APP-Token`	string	Yes	*client* `accessToken`

Body

Parameter name	*Type*	*Required*	*Description*
`accessToken`	string	Yes	*user* `accessToken`
`sceneCode`	string	Yes	Scene code

Response

Parameter name	*Type*	*Description*
`code`	int	Status code
`msg`	string	Status

Response Example

```
{
    "msg": "OK",
    "code": 20000
}
```

Appendix

1. code & msg list

20000=OK

20001=The system is busy, please try again later
20010=The device does not exist
20011=The device does not belong to the current account
20012=Unknown request parameter
20013=Device offline
20100=The user is not bound to the device
20104=Incorrect username or password
20105=The account does not exist
20106=User does not have permission
20107=The username is too long
20108=Token error
20200=Not writable attribute
20300=The scene does not exist
20303=The configuration rules for scene cannot be empty

30001={0} can not be null

30101=Invalid appKey
30102=Incorrect appSecrect
30104=Unauthorized application
30111=Application access token is invalid
30112=Application access token is expired
30113=Application refresh token is invalid
30114=Application refresh token is expired

30211=User access token is invalid
30212=User access token is expired
30213=User refresh token is invalid
30214=User refresh token is expired

30701=IP address is not in the whitelist

99999=Unknown error

2. deviceType list

100	General motors/receivers
101	Tdbu blinds
102	Zebra blinds
201	Bridges
202	Bridges
220	Wi-Fi curtains
221	Wi-Fi receivers
222	Wi-Fi tubular motors
225	Wi-Fi receviers

3. type list

1	Roller Blinds
2	Venetian Blinds
3	Roman Blinds
4	Honeycomb Blinds
5	Shangri-La Blinds
6	Roller Shutter
7	Roller Gate
8	Awning
10	Daynight Blinds
11	Dimming Blinds
12	Curtain
13	Curtain(right stick)
14	Curtain(left stick)