

DEVELOPER REFERENCE · VERSION 1.4.0

V2 API Documentation.

Authentication, device, effect, schedule, group and backup endpoints.

October 2026

Contents.

Version History
1. API Authentication
2. Base URL
3. Get device list
4. Get device detail data
5. Set device switch state
6. Set device name
7. Set device color order
8. Set device IC
9. Set device port
10. Preview built-in effect
11. Preview custom effect
12. Add/Update effect
13. Check out effect
14. Delete effect
15. Update daily schedule
16. Add/Update calendar schedule
17. Delete calendar schedule
18. Set combined effect
19. Get group list
20. Add a new group
21. Update a group
22. Rename a group
23. Delete a group
24. Sync a group
25. Notify update shadow data
26. Update overlay effect
27. Update device datetime
28. Backup device data
29. Get device backup data
30. Restore device
Appendix

Version History

Version	Description	Date
1.0.0	Initial version.	2021-04-20
1.1.0	Add currentEffect data to the device details response data.	2021-11-22
1.2.0	Add group function APIs. (19 – 23)	2021-12-04
1.2.1	Add the “sync a group” API. (24)	2021-12-08
1.3.0	Add the “Notify update shadow data” API. (25)	2022-08-18
1.4.0	1. Add “Overlay effect” API. (26) 2. Add “Update device datetime” API. (27) 3. Add APIs for backing up device data. (28 – 30) 4. Add “overlayEffects” and “currentDatetime” to the device details.	2024-09-26

1. API Authentication

All API requests must include authentication headers.

1.1 Request Headers

Each request must include the following HTTP headers:

```
Authorization: <accessToken>
S-ClientId: <clientId>
S-Timestamp: <timestamp>
```

Header	Required	Description
`Authorization`	Yes	The access token generated using HMAC-SHA256
`S-ClientId`	Yes	Unique client identifier
`S-Timestamp`	Yes	Current timestamp in milliseconds since Unix epoch

1.2 How to Generate accessToken

The accessToken must be generated using the following steps:

Step 1: Concatenate the string. Use the following format:

```
Trimlight|<clientId>|<timestamp>
```

Step 2: Compute HMAC-SHA256. Use clientSecret as the key to compute the HMAC-SHA256 hash of the string from Step 1.

Step 3: Base64 encode. Encode the result from Step 2 using Base64. The output is the final `accessToken`.

1.3 Example

Given:

- clientId: `tester`
- clientSecret: `test_secret`
- timestamp: `1713166849256`

Concatenated string:

```
Trimlight|tester|1713166849256
```

Generated accessToken:

```
z02N77XySuOwv5OSUe0vrPwprRITb656xKPulS9ooXI=
```

This value should be used as the Authorization header.

1.4 Full Request Example

A Python example for getting the device list:

```python
import http.client

conn = http.client.HTTPSConnection("trimlight.ledhue.com")

payload = ""

headers = {
    'Content-Type': "application/json",
    'Authorization': "z02N77XySuOwv5OSUe0vrPwprRITb656xKPulS9ooXI=",
    'S-ClientId': "tester",
    'S-Timestamp': "1713166849256"
    }

conn.request("GET", "/trimlight/v1/oauth/resources/devices", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

1.5 Notes

- The timestamp must be in milliseconds.
- Keep your clientSecret secure and do not expose it.
- It is recommended to generate a new accessToken for each request.
- Requests may be rejected if the timestamp is outside the allowed time window (to prevent replay attacks).

1.6 Credentials

Please contact our business team to obtain:

- clientId
- clientSecret

2. Base URL

POST `https://trimlight.ledhue.com/trimlight`

3. Get device list

GET `/v1/oauth/resources/devices`

Request params:

```
{
    "page": 1  // 10 devices on one page
               // If the value is 0 or null, it will return to the list of all devices
}
```

Response result:

```
{
    "code": 0,
    "desc": "success",
    "payload": {
        "total": 2,
        "current": 1,
        "data": [
            {
                "deviceId": "xxxxxxxxxxx1",
                "name": "xxxx1",
                "switchState": 0,
                "connectivity": 1,
                "state": 0,
                "fwVersionName": "1.1.1"
            },
            {
```

```
                    "deviceId": "xxxxxxxxxx2",
                    "name": "xxxx2",
                    "switchState": 0,
                    "connectivity": 1,
                    "state": 0,
                    "fwVersionName": "1.1.1"
                }
            ]
        }
    }
```

Response result code details

Field	Description	Type
code	Result code. [1]	Integer
desc	Result description.	String
payload	Result payload.	Object

Page fields details

Field	Description	Type
total	Number of all devices.	Integer
page	The page number of the current data.	Integer
data	Device list.	List

Device fields details

Field	Description	Type
deviceId	Unique ID of the device.	String
name	Device name.	String
switchState	Device switch state. 0 : light off. 1 : manual mode. 2 : timer mode.	Integer
connectivity	Device connectivity state. 0 : offline. 1 : online.	Integer
state	Device state. 0 : normal. 1 : upgrading.	Integer
fwVersionName	Device firmware version name.	String

4. Get device detail data

`POST` `/v1/oauth/resources/device/get`

Request params:

```
    {
        "deviceId": "<device-id>",
        "currentDate": {
            "year": 21, // 2021
            "month": 1,
            "day": 1,
            "weekday": 1,
            "hours": 1,
            "minutes": 1,
            "seconds": 1
        }
    }
```

Current date details

Field	Description	Type

year	Current date year. Years relative to 2000.	Integer
month	Current date month. Range: [1, 12].	Integer
day	Current date day. Range: [1, 31].	Integer
weekday	Current day of week. SUNDAY = 1, MONDAY = 2, TUESDAY = 3, WEDNESDAY = 4, THURSDAY = 5, FRIDAY = 6, SATURDAY = 7	Integer
hours	Current date year. Range: [0, 23].	Integer
minutes	Current date year. Range: [0, 59].	Integer
seconds	Current date year. Range: [0, 59].	Integer

Response result:


```
{
    "code": 0,
    "desc": "success",
    "payload": {
        "name": "xxxx2",
        "switchState": 0,
        "connectivity": 1,
        "state": 0,
        "colorOrder": 0,
        "ic": 0,
        "ports": [
            { "id": 0, "start": 1, "end": 1024 },
            { "id": 1, "start": 1, "end": 1024 },
            { "id": 2, "start": 1, "end": 1024 },
            { "id": 3, "start": 1, "end": 1024 }
        ],
        "fwVersionName": "1.1.1",
        "effects": [
            {
                "id": 0,
                "name": "New Year",
                "category": 0,
                "mode": 0,
                "speed": 100,
                "brightness": 100,
                "pixelLen": 30,
                "reverse": false
            },
            {
                "id": 1,
                "name": "xxxxxxxx2",
                "category": 1,
                "mode": 0,
                "speed": 100,
                "brightness": 100,
                "pixels": [
                    { "index": 0, "count": 5,  "color": 16711680, "disable": false },
                    { "index": 1, "count": 10, "color": 65280,    "disable": false },
                    { "index": 2, "count": 10, "color": 255,      "disable": false }
                ]
            }
        ],
        "combinedEffect": {
            "effectIds": [0, 2, 3],
            "interval": 5
        },
        "daily": [
            {
                "id": 0,
                "enable": true,
                "effectId": 0,
                "repetition": 1,
                "startTime": { "hours": 10, "minutes": 1 },
                "endTime":   { "hours": 11, "minutes": 1 }
            },
            {
                "id": 1,
                "enable": true,
```

```
                        "effectId": 1,
                        "repetition": 1,
                        "startTime": { "hours": 10, "minutes": 1 },
                        "endTime":   { "hours": 11, "minutes": 1 }
                    }
                ],
                "calendar": [
                    {
                        "id": 0,
                        "effectId": 1,
                        "startDate": { "month": 12, "day": 31 },
                        "endDate":   { "month": 1,  "day": 1 },
                        "startTime": { "hours": 10, "minutes": 1 },
                        "endTime":   { "hours": 11, "minutes": 1 }
                    },
                    {
                        "id": 1,
                        "effectId": 2,
                        "startDate": { "month": 12, "day": 31 },
                        "endDate":   { "month": 1,  "day": 1 },
                        "startTime": { "hours": 10, "minutes": 1 },
                        "endTime":   { "hours": 11, "minutes": 1 }
                    }
                ],
                "currentEffect": {
                    "category": 1,
                    "mode": 1,
                    "speed": 174,
                    "brightness": 204,
                    "pixelLen": 37,
                    "reverse": false
                },
                "overlayEffects": [
                    {
                        "overlayType": 1,
                        "targetEffect": 6
                    }
                ],
                "currentDatetime": {
                    "year": 24,
                    "month": 9,
                    "day": 26,
                    "weekday": 4,
                    "hours": 16,
                    "minutes": 20,
                    "seconds": 2
                }
            }
        }
```


4.1 Device fields details

Field	Description	Type
name	Device name.	String
switchState	Device switch state. 0 : light off. 1 : manual mode. 2 : timer mode.	Integer
connectivity	Device connectivity state. 0 : offline. 1 : online.	Integer
state	Device state. 0 : normal. 1 : upgrading.	Integer
colorOrder	Color order. [2]	Integer
ic	IC. [3]	Integer
ports	The pixel setting for each port. id: Port ID, range from 0 to 3, corresponds to port1, port2, port3, port4. start: The start pixel of the port. Range: [1, 2048]. end: The end pixel of the port. Range: [1, 2048]. “start” should not be greater than “end”.	List
fwVersionName		

	Device firmware version name.	String
`effects`	All the effects stored in the device.	List
`combinedEffect`	Combined effect.	Object
`daily`	Daily schedules. Each device has two daily schedules.	List
`calendar`	Calendar schedules.	List
`currentEffect`	Device current running effect. (If the device's switch state is timer mode, although the light is off at this time, it will return the last running effect data.) Note: The effect ID will be -1 when the controller is running a preview effect (not yet saved).	Object
`overlayEffects`	All overlay effects.	List
`currentDatetime`	The current date and time of the device.	Object

4.2 Effect fields details

Field	Description	Type
`id`	Effect ID. ID of the saved effect. Note that it may be -1 in currentEffect when the controller is running a preview effect (not yet saved).	Integer
`category`	Effect category. 0 : built-in effect. 1 : custom effect.	Integer
`mode`	Effect mode. Built-in effect (category value is 0) mode range: [0, 179]. [4] Custom effect (category value is 1) mode range: [0, 16]. [5]	Integer
`speed`	Effect speed. Speed range: [0, 255].	Integer
`brightness`	Effect brightness. Brightness range: [0, 255].	Integer
`pixelLen`	Effect pixel length. (Only required for built-in effects) Pixel length range: [1, 90].	Integer
`reverse`	Reverse effect. (Only required for built-in effects)	Boolean
`pixels`	Custom effect pixels. (Only required for built-in effects)	List

4.3 Pixel fields details

Field	Description	Type
`index`	Custom effect pixel index.	Integer
`count`	Custom effect pixel count.	Integer
`color`	Custom effect pixel color.	Integer
`disable`	Disable pixel.	Boolean

4.4 Combined effect fields details

Field	Description	Type
`effectIds`	A list of each effect's ID in the combined effect.	List
`interval`	The interval between switching to the next effect. (Unit: minute.)	Integer

4.5 Daily schedule fields details

Field	Description	Type
`id`	Daily schedule ID. 0: daily schedule 1. 1: daily schedule 2.	List
`enable`	Enable daily schedule.	Boolean
`effectId`	Effect ID in the daily schedule.	Integer
`repetition`	The repetition of daily schedule. [6]	Integer
`startTime`	The start time of daily schedule. See schedule time fields details.	Object
`endTime`	The end time of daily schedule. See schedule time fields details.	Object

4.6 Schedule time fields details

Field	Description	Type
`hours`	Schedule hours, range from [1, 24].	Integer
`minutes`	Schedule minutes, range from [1, 60].	Integer

4.7 Calendar schedule fields details

Field	Description	Type
`id`	Calendar schedule ID. Schedule ID range: [0, 59]. Up to 60 calendar schedules can be saved.	List
`effectId`	Effect ID in the calendar schedule.	Integer
`startDate`	The start date of calendar schedule. See schedule date fields details.	Object
`endDate`	The end date of calendar schedule. See schedule date fields details.	Object
`startTime`	The start time in each day of calendar schedule. See schedule time fields details.	Object
`endTime`	The end time in each day of calendar schedule. See schedule time fields details.	Object

4.8 Schedule date fields details

Field	Description	Type
`month`	Schedule month, range from [1, 12].	Integer
`day`	Schedule day, range from [1, 31].	Integer

4.9 Current effect fields details

Field	Description	Type
`category`	Effect category. 0 : built-in effect. 1 : custom effect.	Integer
`mode`	Effect mode. Built-in effect (category value is 0) mode range: [0, 179]. [4] Custom effect (category value is 1) mode range: [0, 16]. [5]	Integer
`speed`	Effect speed. Speed range: [0, 255].	Integer
`brightness`	Effect brightness. Brightness range: [0, 255].	Integer

`pixelLen`	Effect pixel length. (Only available for built-in effects) Pixel length range: [1, 90].	Integer
`reverse`	Reverse effect. (Only available for built-in effects)	Boolean
`pixels`	Custom effect pixels. (Only available for built-in effects)	List

4.10 Overlay effect fields details

Field	Description	Type
`overlayType`	Overlay effect type. 0 : Lightning. 1 : Snow.	Integer
`targetEffect`	The target effect ID.	Integer

Overlay Type

Name	Value
`Lightning`	0
`Snow`	1

5. Set device switch state

`POST` `/v1/oauth/resources/device/update`

Request params:

```
{
    "deviceId": "<device-id>",
    "payload": {
        "switchState": 0
    }
}
```

Response result:

```
{
    "code": 0,
    "desc": "success"
}
```

6. Set device name

`POST` `/v1/oauth/resources/device/update`

Request params:

```
{
    "deviceId": "<device-id>",
    "payload": {
        "name": "xxx"
    }
}
```

Response result:

```
{
    "code": 0,
    "desc": "success"
}
```

7. Set device color order

`POST` `/v1/oauth/resources/device/update`

Request params:

```
{
    "deviceId": "<device-id>",
    "payload": {
        "colorOrder": 0
    }
}
```

Response result:

```
{
    "code": 0,
    "desc": "success"
}
```

8. Set device IC

`POST` `/v1/oauth/resources/device/update`

Request params:

```
{
    "deviceId": "<device-id>",
    "payload": {
        "ic": 0
    }
}
```

Response result:

```
{
    "code": 0,
    "desc": "success"
}
```

9. Set device port

`POST` `/v1/oauth/resources/device/update`

Request params:

```
{
    "deviceId": "<device-id>",
    "payload": {
        "ports": [
            {
                "id": 0, // [0, 3]
                "start": 1,
                "end": 1024
            }
        ]
    }
}
```

Response result:

```
{
    "code": 0,
    "desc": "success"
}
```

10. Preview built-in effect

`POST` `/v1/oauth/resources/device/effect/preview`

Request params:

```
{
    "deviceId": "<device-id>",
    "payload": {
        "category": 0,
        "mode": 0,
        "speed": 100,
        "brightness": 100,
        "pixelLen": 10,
        "reverse": false
    }
}
```

Effect fields details

Field	Description	Type
`category`	Effect category. 0 : built-in effect. 1 : custom effect. Here the value is 0.	Integer
`mode`	Built-in effect mode. Built-in effect mode range: [0, 179]. [4]	Integer
`speed`	Effect speed. Speed range: [0, 255].	Integer
`brightness`	Effect brightness. Brightness range: [0, 255].	Integer
`pixelLen`	Effect pixel length. Pixel length range: [1, 90].	Integer
`reverse`	Reverse effect.	Boolean

Response result:

```
{
    "code": 0,
    "desc": "success"
}
```

Example — Built-in effect (as shown in the source)

```
{
    "deviceId": "<device-id>",
    "payload": {
        "category": 1,
        "mode": 1,
        "speed": 100,
        "brightness": 100,
        "pixelLen": 10,
        "reverse": false
    }
}
```

11. Preview custom effect

`POST` `/v1/oauth/resources/device/effect/preview`

Request params:

```
{
    "deviceId": "<device-id>",
    "payload": {
        "category": 1,
```

```
                "mode": 0,
                "speed": 100,
                "brightness": 100,
                "pixels": [
                    {
                        "index": 0,
                        "count": 5,
                        "color": 16711680, // (0xFF0000)
                        "disable": false
                    },
                    {
                        "index": 1,
                        "count": 10,
                        "color": 16711680,
                        "disable": false
                    }
                ]
            }
        }
```

Effect fields details

Field	Description	Type
`category`	Effect category. 0 : built-in effect. 1 : custom effect. Here the value is 1.	Integer
`mode`	Effect mode. Custom effect mode range: [0, 16]. [5]	Integer
`speed`	Effect speed. Speed range: [0, 255].	Integer
`brightness`	Effect brightness. Brightness range: [0, 255].	Integer
`pixels`	Custom effect pixels.	List

Pixel fields details

Field	Description	Type
`index`	Custom pixel index. Index range: [0, 29].	Integer
`count`	Custom pixel count. Pixel count range: [0, 60].	Integer
`color`	Custom pixel color decimal value. e.g. 0xFF0000 => 16711680.	Integer
`disable`	Custom pixel disable.	Boolean

Example — Custom effect (as shown in the source)

```
        {
            "deviceId": "<device-id>",
            "payload": {
                "category": 2,
                "mode": 1,
                "speed": 100,
                "brightness": 100,
                "pixels": [
                    {
                        "index": 0,
                        "count": 10,
                        "color": 255,
                        "disable": false
                    },
                    {
                        "index": 1,
                        "count": 2,
                        "color": 0,
                        "disable": true
                    },
                    {
                        "index": 2,
                        "count": 3,
                        "color": 65280,
```

```
                    "disable": false
                },
                {
                    "index": 2, // The same index will be overwritten
                    "count": 3,
                    "color": 16711680,
                    "disable": false
                }
            ]
        }
    }
```

12. Add/Update effect

POST `/v1/oauth/resources/device/effect/save`

Request params:

```
    {
        "deviceId": "<device-id>",
        "payload": {
            "id": -1
            "name": "xxxx",
            "category": 1/2,
            "mode": 0,
            "speed": 100,
            "brightness": 100,
            "pixelLen": 10,
            "reverse": false,
            "pixels": [
                {
                    "index": 0,
                    "count": 5,
                    "color": 16711680, // (0xFF0000)
                    "disable": false
                },
                {
                    "index": 1,
                    "count": 10,
                    "color": 16711680,
                    "disable": false
                }
            ]
        }
    }
```

Effect fields details

Field	Description	Type
`id`	Effect ID. If ID is -1 or null, it will be saved as a new effect; if not, the effect corresponding to the ID will be updated. Up to 60 effects can be saved.	Integer
`category`	Effect category. 0 : built-in effect. 1 : custom effect.	Integer
`mode`	Effect mode. Custom effect mode range: [0, 16]. [5]	Integer
`speed`	Effect speed. Speed range: [0, 255].	Integer
`brightness`	Effect brightness. Brightness range: [0, 255].	Integer
`pixelLen`	Effect pixel length. Pixel length range: [1, 90]. (Only required for built-in effects)	Integer
`reverse`	Reverse effect. (Only required for built-in effects)	Boolean
`pixels`	Custom effect pixels. See pixel fields details.	List

(Only required for custom effects)

Response result:

```
{
    "code": 0,
    "desc": "success",
    "payload": {
        "id": 10   // effect id
    }
}
```

13. Check out effect

`POST` `/v1/oauth/resources/device/effect/view`

Request params:

```
{
    "deviceId": "<device-id>",
    "payload": {
        "id": 0
    }
}
```

Response result:

```
{
    "code": 0,
    "desc": "success"
}
```

14. Delete effect

`POST` `/v1/oauth/resources/device/effect/delete`

Request params:

```
{
    "deviceId": "<device-id>",
    "payload": {
        "id": 0
    }
}
```

Response result:

```
{
    "code": 0,
    "desc": "success"
}
```

15. Update daily schedule

`POST` `/v1/oauth/resources/device/daily/save`

Request params:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "id": 1,
    "enable": true,
    "effectId": 1,
    "repetition": 1,
```

```
      "startTime": {
        "hours": 10,
        "minutes": 1
      },
      "endTime": {
        "hours": 11,
        "minutes": 1
      },
      "currentDate": {
        "month": 1,
        "day": 1
      }
    }
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

16. Add/Update calendar schedule

POST /v1/oauth/resources/device/calendar/save

Request params:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "id": 0,
    "effectId": 1, // The combined effect id is fixed at 200
    "startDate": {
      "month": 12,
      "day": 31
    },
    "endDate": {
      "month": 1,
      "day": 1
    },
    "startTime": {
      "hours": 10,
      "minutes": 1
    },
    "endTime": {
      "hours": 11,
      "minutes": 1
    }
  }
}
```

Response result:

```
{
  "code": 0,
  "desc": "success",
  "payload": {
    "id": 10   // calendar id
  }
}
```

17. Delete calendar schedule

POST /v1/oauth/resources/device/calendar/delete

Request params:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "id": 0
```

```
  }
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

18. Set combined effect

POST `/v1/oauth/resources/device/combined-effect/save`

Request params:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "effectIds": [0, 2, 3], // up to 5 effects
    "interval": 5
  }
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

19. Get group list

GET `/v1/oauth/resources/groups`

Request params:

```
{
  "page": 1  // 10 groups on one page
             // If the value is 0 or null, it will return to the list of all groups
}
```

Response result:

```
{
    "code": 0,
    "desc": "success",
    "payload": {
        "total": 2,
        "current": 1,
        "data": [
            {
                "groupId": "xxx",
                "name": "group1",
                "masterDevice": {
                    "deviceId": "<device_id>",
                    "name": "<device_name>"
                },
                "devices": [
                    {
                        "deviceId": "<device_id>",
                        "name": "<device_name>"
                    }
                ]
            },
            {
                "groupId": "xxx",
                "name": "group2",
                "masterDevice": {
                    "deviceId": "<device_id>",
```

```
                        "name": "<device_name>"
                    },
                    "devices": [
                        {
                            "deviceId": "<device_id>",
                            "name": "<device_name>"
                        }
                    ]
                }
            ]
        }
    }
```

20. Add a new group

GET /v1/oauth/resources/group/add

Request params:

```
    {
      "name": "<new group name>",
      "masterDevice": "<master device ID>",
      "devices": [
        "<master device ID>",
        "<device ID>",
        "<device ID>"
      ]
    }
```

Response result:

```
    {
        "code": 0,
        "desc": "success",
        "payload": {
            "groupId": "<new group ID>"
        }
    }
```

21. Update a group

GET /v1/oauth/resources/group/update

Request params:

```
    {
      "groupId": "<group ID>",
      "masterDevice": "<master device ID>",
      "devices": [
        "<master device ID>",
        "<device ID>",
        "<device ID>"
      ]
    }
```

Response result:

```
    {
        "code": 0,
        "desc": "success",
        "payload": {
            "groupId": "<new group ID>"
        }
    }
```

22. Rename a group

GET /v1/oauth/resources/group/rename

Request params:

```
{
  "groupId": "<group ID>",
  "name": "<new group name>"
}
```

Response result:

```
{
    "code": 0,
    "desc": "success"
}
```

23. Delete a group

`GET` `/v1/oauth/resources/group/delete`

Request params:

```
{
  "groupId": "<group ID>"
}
```

Response result:

```
{
    "code": 0,
    "desc": "success"
}
```

24. Sync a group

`GET` `/v1/oauth/resources/group/sync`

Request params:

```
{
  "groupId": "<group ID>"
}
```

Response result:

```
{
    "code": 0,
    "desc": "success"
}
```

PS: The master device must be online.

25. Notify update shadow data

Before requesting detailed data for the device, you can send this request to notify the device to report the latest shadow data.

`GET` `/v1/oauth/resources/device/notify-update-shadow`

Request params:

```
{
  "deviceId": "<device-id>",
  "currentDate": {
    "year": 21, // 2021
```

```
            "month": 1,
            "day": 1,
            "weekday": 1,
            "hours": 1,
            "minutes": 1,
            "seconds": 1
        }
    }
```

Response result:

```
    {
        "code": 0,
        "desc": "success"
    }
```

26. Update overlay effect

Add specific overlay effect to effects.

POST `/v1/oauth/resources/device/effect/overlay`

Request params:

```
    {
        "deviceId": "<device-id>",
        "payload": {
            "overlayEffects": [
                {
                    "overlayType": <overlayTypeValue>,
                    "targetEffect": <effectId>
                },
                {
                    "overlayType": <overlayTypeValue>,
                    "targetEffect": <effectId>
                },
                ...
            ]
        }
    }
```

If you want to remove all overlay effects, “overlayEffects” can be null or an empty list.

Overlay Type

Name	Value
Lightning	0
Snow	1

Response result:

```
    {
        "code": 0,
        "desc": "success"
    }
```

27. Update device datetime

POST `/v1/oauth/resources/device/datetime/sync`

Request params:

```
    {
      "deviceId": "<device-id>",
      "currentDate": {
        "year": 21, // 2021
        "month": 1,
        "day": 1,
        "weekday": 1,
        "hours": 1,
```

```
            "minutes": 1,
            "seconds": 1
          }
        }
```

Response result:

```
        {
            "code": 0,
            "desc": "success"
        }
```

28. Backup device data

POST /v1/oauth/resources/device/data/backup

Request params:

```
        {
          "deviceId": "<device-id>",
          "payload": {
            "backupTime": "yyyy-MM-dd HH:mm:ss timezone" // current date and time, e.g. "2024-09-26 16:20:26 PDT"
          }
        }
```

timezone: `(HST|AKST|PST|MST|CST|EST|HDT|AKDT|PDT|MDT|CDT|EDT)`

Response result:

```
        {
            "code": 0,
            "desc": "success"
        }
```

29. Get device backup data

POST /v1/oauth/resources/device/data/backup/get

Request params:

```
        {
          "deviceId": "<device-id>"
        }
```

Response result:

```
        {
            "code": 0,
            "desc": "success",
            "payload": {
                "backupKey": "xxx",
                "backupTime": "2024-09-26 16:20:26 PDT"
            }
        }
```

30. Restore device

POST /v1/oauth/resources/device/data/backup/restore

Request params:

```
        {
          "deviceId": "<device-id>",
          "payload": {
            "backupKey": "xxx"
```

```
        }
    }
```

Response result:

```
    {
        "code": 0,
        "desc": "success"
    }
```

Appendix

[1] Result code

Code	Description
0	Success.
10001	Error.
10002	Wrong password.

[2] Color order

Color order	Value
RGB	0
RBG	1
GRB	2
GBR	3
BRG	4
BGR	5

[3] IC

IC	Value
UCS1903	0
DMX512	1

[4] Built-in effect mode

Value	Effect	Value	Effect	Value	Effect
0	Rainbow Gradual Chase	60	Yellow Cyan Wave	120	Blue White Gradual Snake
1	Rainbow Comet	61	Yellow Purple Wave	121	Yellow Cyan Gradual Snake
2	Rainbow Segment	62	Yellow White Wave	122	Yellow Purple Gradual Snake
3	Rainbow Wave	63	Cyan Purple Wave	123	Yellow White Gradual Snake
4	Rainbow Meteor	64	Cyan White Wave	124	Cyan Purple Gradual Snake
5	Rainbow Gradual	65	Purple White Wave	125	Cyan White Gradual Snake
6	Rainbow Jump	66	Red Dot Pulse	126	Purple White Gradual Snake
7	Rainbow Stars	67	Green Dot Pulse	127	Red White Blank Snake
8	Rainbow Fade In Out	68	Blue Dot Pulse	128	Green White Blank Snake
9	Rainbow Spin	69	Yellow Dot Pulse	129	Blue White Blank Snake
10	Red Stacking	70	Cyan Dot Pulse	130	Yellow White Blank Snake
11	Green Stacking	71	Purple Dot Pulse	131	Cyan White Blank Snake
12	Blue Stacking	72	White Dot Pulse	132	Purple White Blank Snake
13	Yellow Stacking	73	Red Green Blank Pulse	133	Green Yellow White Snake
14	Cyan Stacking	74	Green Blue Blank Pulse	134	Red Green White Snake
15	Purple Stacking	75	Blue Yellow Blank Pulse	135	Red Yellow Snake
16	White Stacking	76	Yellow Cyan Blank Pulse	136	Red White Snake
17	Full Color Stack	77	Cyan Purple Blank Pulse	137	Green White Snake
18	Red to Green Stack	78	Purple White Blank Pulse	138	Red Stars

19	Green to Blue Stack	79	Red with Purple Pulse	139	Green Stars
20	Blue to Yellow Stack	80	Green with Cyan Pulse	140	Blue Stars
21	Yellow to Cyan Stack	81	Blue with Yellow Pulse	141	Yellow Stars
22	Cyan to Purple Stack	82	Yellow with Blue Pulse	142	Cyan Stars
23	Purple to White Stack	83	Cyan with Green Pulse	143	Purple Stars
24	Red Comet	84	Purple with Purple Pulse	144	White Stars
25	Green Comet	85	Red Comet Spin	145	Red Background Stars
26	Blue Comet	86	Green Comet Spin	146	Green Background Stars
27	Yellow Comet	87	Blue Comet Spin	147	Blue Background Stars
28	Cyan Comet	88	Yellow Comet Spin	148	Yellow Background Stars
29	Purple Comet	89	Cyan Comet Spin	149	Cyan Background Stars
30	White Comet	90	Purple Comet Spin	150	Purple Background Stars
31	Red Meteor	91	White Comet Spin	151	Red White Background Stars
32	Green Meteor	92	Red Dot Spin	152	Green White Background Stars
33	Blue Meteor	93	Green Dot Spin	153	Blue White Background Stars
34	Yellow Meteor	94	Blue Dot Spin	154	Yellow White Background Stars
35	Cyan Meteor	95	Yellow Dot Spin	155	Cyan White Background Stars
36	Purple Meteor	96	Cyan Dot Spin	156	Purple White Background Stars
37	White Meteor	97	Purple Dot Spin	157	White White Background Stars
38	Red Wave	98	White Dot Spin	158	Red Breath
39	Green Wave	99	Red Segment Spin	159	Green Breath
40	Blue Wave	100	Green Segment Spin	160	Blue Breath
41	Yellow Wave	101	Blue Segment Spin	161	Yellow Breath
42	Cyan Wave	102	Yellow Segment Spin	162	Cyan Breath
43	Purple Wave	103	Cyan Segment Spin	163	Purple Breath
44	White Wave	104	Purple Segment Spin	164	White Breath
45	Red Green Wave	105	White Segment Spin	165	Red Yellow Fire
46	Red Blue Wave	106	Red Green Gradual Snake	166	Red Purple Fire
47	Red Yellow Wave	107	Red Blue Gradual Snake	167	Green Yellow Fire
48	Red Cyan Wave	108	Red Yellow Gradual Snake	168	Green Cyan Fire
49	Red Purple Wave	109	Red Cyan Gradual Snake	169	Blue Purple Fire
50	Red White Wave	110	Red Purple Gradual Snake	170	Blue Cyan Fire
51	Green Blue Wave	111	Red White Gradual Snake	171	Red Strobe
52	Green Yellow Wave	112	Green Blue Gradual Snake	172	Green Strobe
53	Green Cyan Wave	113	Green Yellow Gradual Snake	173	Blue Strobe
54	Green Purple Wave	114	Green Cyan Gradual Snake	174	Yellow Strobe
55	Green White Wave	115	Green Purple Gradual Snake	175	Cyan Strobe
56	Blue Yellow Wave	116	Green White Gradual Snake	176	Purple Strobe
57	Blue Cyan Wave	117	Blue Yellow Gradual Snake	177	White Strobe
58	Blue Purple Wave	118	Blue Cyan Gradual Snake	178	Red Blue White Strobe
59	Blue White Wave	119	Blue Purple Gradual Snake	179	Full Color Strobe

[5] Custom effect mode

Mode	Value
STATIC	0
CHASE FORWARD	1
CHASE BACKWARD	2
CHASE MIDDLE TO OUT	3
CHASE OUT TO MIDDLE	4
STARS	5
BREATH	6
COMET FORWARD	7
COMET BACKWARD	8
COMET MIDDLE TO OUT	9
COMET OUT TO MIDDLE	10

WAVE FORWARD	11
WAVE BACKWARD	12
WAVE MIDDLE TO OUT	13
WAVE OUT TO MIDDLE	14
STROBE	15
SOLID FADE	16

[6] Repetition

Repetition	Value
Today Only	0
Everyday	1
Week Days	2
Weekend	3