

Hunter Douglas PowerView® Hub API for Home Automation Integration

Appendix document

by

Jacob Laursen

`jacob at vindvejr dot dk`

Table of Contents

Overview	5
Document Conventions	5
Version information	5
Scene Collections.....	5
Scheduled Events	10
Definitions.....	16
EventType.....	16
Name	16
SceneCollection	16
SceneCollectionId	16
SceneCollectionObject.....	16
SceneCollectionsResponse	17
ScheduledEvent	17
ScheduledEventObject	18
ScheduledEventsResponse	18
UniqueId	18

Overview

This document is an appendix to the document **PowerView-Hub-REST-API-v2.pdf** which describes the PowerView® Hub REST API. The source of the original document is unknown, but it seems it might have been published by Hunter Douglas.

This appendix document describes some methods which are missing in the original document, or not fully described. This is based on reverse engineering by intercepting the communication between the PowerView app and the PowerView Hub.

Document Conventions

See original document.

Version information

Version: 1.0.0

Scene Collections

Scene collections are no longer supported by the PowerView app. To be precise, creating new scene collections is no longer supported, but editing and deleting existing scene collections is still possible (for now). They are still fully supported by the PowerView Hub (for now).

Get all scene collections

Please see original document. Important note, though: Hub v2 redirects `/api/scenecollections` to `/api/sceneCollections`. The latter can also be called directly to avoid redirection, but Hub v1 only supports the former:

```
GET /api/scenecollections
```

Create a scene collection

```
POST /api/scenecollections/
```

Example HTTP request

Request path

```
POST /api/scenecollections/
```

Request body

```
{
  "sceneCollection": {
    "name": "Qs04cm4gaSBzZW5n",
    "colorId": 12,
    "iconId": 17,
    "id": -1,
    "order": 0,
    "hkAssist": false
  }
}
```

Response 200

```
{
  "sceneCollection": {
    "name": "Qs04cm4gaSBzZW5n",
    "colorId": 12,
    "iconId": 17,
    "id": 16820,
    "order": 0,
    "hkAssist": false
  }
}
```

An empty scene collection has now been created. The returned id can be used to include existing scenes in the scene collection.

Modify a scene collection

```
PUT /api/scenecollections/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	Unique id of the scene collection.	Integer

Example HTTP request

Request path

```
PUT /api/scenecollections/16820
```

Request body

```
{
  "sceneCollection": {
    "name": "Qs04cm4gaSBzZW5n",
    "colorId": 12,
    "iconId": 17,
    "id": 16820,
    "order": 0,
    "hkAssist": false
  }
}
```

Add a scene to a scene collection

```
POST /api/scenecollectionmembers/
```

Request body

```
{
  "sceneCollectionMember": {
    "sceneCollectionId": 16820,
    "sceneId": 19165
  }
}
```

At least two scenes are needed for the scene collection to make sense.

Delete a scene from a scene collection

```
DELETE
/api/scenecollectionmembers?sceneCollectionId={sceneCollectionId}&sceneId={sceneId}
```

Parameters

Type	Name	Description	Schema
Path	sceneCollectionId <i>required</i>	Unique id of the scene collection.	Integer
Path	sceneId <i>required</i>	Unique id of the scene.	Integer

Responses

HTTP Code	Description	Schema
200	Scene collection member deleted.	
400	Bad client request.	No Content
404	Resource not found.	No Content
423	Hub is temporarily busy for maintenance (Always returns UserData).	No Content
500	Internal server error.	No Content

Example HTTP request

Request path

```
DELETE /api/scenecollectionmembers?sceneCollectionId=16328&sceneId=19165
```

Delete a scene collection

```
DELETE /api/scenecollections/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	Unique id of the scene collection.	Integer

Responses

HTTP Code	Description	Schema
204	Scene collection deleted.	No Content
400	Bad client request.	No Content
404	Resource not found.	No Content

423	Hub is temporarily busy for maintenance (Always returns UserData).	No Content
500	Internal server error.	No Content

Example HTTP request

Request path

```
DELETE /api/scenecollections/16820
```

Scheduled Events

Get all scheduled events

```
GET /api/scheduleevents/
```

Description

- Gets a list of all scheduled event ids and the corresponding scheduled event data.
- The scheduled event data is returned in the same order as the scheduled event ids.
- If no scheduled events exist, then empty arrays for scheduled event ids and scheduled event data are returned.

Responses

HTTP Code	Description	Schema
200	Scheduled events returned.	ScheduledEventsResponse
400	Bad client request.	No Content
423	Hub is temporarily busy for maintenance (Always returns UserData).	No Content
500	Internal server error.	No Content

Example HTTP request

Request path

```
/api/scheduleevents
```

Example HTTP response

Response 200


```

{
  "scheduledEventIds": [
    44234
  ],
  "scheduledEventData": [
    {
      "enabled": true,
      "sceneId": 7829,
      "daySunday": true,
      "dayMonday": true,
      "dayTuesday": true,
      "dayWednesday": true,
      "dayThursday": true,
      "dayFriday": true,
      "daySaturday": true,
      "eventType": 2,
      "hour": 0,
      "minute": 0,
      "id": 44234
    }
  ]
}

```

Get a scheduled event

GET /api/scheduleevents/{id}

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	Unique id of the scheduled event.	Integer

Responses

HTTP Code	Description	Schema
200	Scheduled event returned.	ScheduledEventObject
400	Bad client request.	No Content
404	Resource not found.	No Content
423	Hub is temporarily busy for maintenance (Always returns UserData).	No Content

500	Internal server error.	No Content
-----	------------------------	------------

Example HTTP request

Request path

```
/api/scheduleevent/44234
```

Example HTTP response

Response 200

```
{
  "scheduledEvent": {
    "enabled": true,
    "sceneId": 7829,
    "daySunday": true,
    "dayMonday": true,
    "dayTuesday": true,
    "dayWednesday": true,
    "dayThursday": true,
    "dayFriday": true,
    "daySaturday": true,
    "eventType": 2,
    "hour": 0,
    "minute": 0,
    "id": 44234
  }
}
```

Update a scheduled event

```
PUT /api/scheduleevents/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	Unique id of the scheduled event.	Integer

Responses

HTTP Code	Description	Schema
200	Scheduled event updated.	ScheduledEventObject

400	Bad client request.	No Content
404	Resource not found.	No Content
423	Hub is temporarily busy for maintenance (Always returns UserData).	No Content
500	Internal server error.	No Content

Example HTTP request

Request path

```
/api/scheduleevent/44234
```

Example HTTP response

Request body

```
{
  "scheduledEvent": {
    "enabled": false,
    "sceneId": 7829,
    "daySunday": true,
    "dayMonday": true,
    "dayTuesday": true,
    "dayWednesday": true,
    "dayThursday": true,
    "dayFriday": true,
    "daySaturday": true,
    "eventType": 2,
    "hour": 0,
    "minute": 0,
    "id": 44234
  }
}
```

Response 200

```
{
  "scheduledEvent": {
    "enabled": false,
    "sceneId": 7829,
    "daySunday": true,
    "dayMonday": true,
    "dayTuesday": true,
    "dayWednesday": true,
    "dayThursday": true,
    "dayFriday": true,
    "daySaturday": true,
    "eventType": 2,
    "hour": 0,
    "minute": 0,
    "id": 44234
  }
}
```


Definitions

EventType

0 = Sunrise, 1 = Sunset, 2 = Time

Type : enum (0, 1, 2)

Name

Base64 encoded name.

Type : string (byte)

SceneCollection

Name	Description	Schema
colorId <i>required</i>	Example : ColorId	ColorId
iconId <i>required</i>	Example : IconId	IconId
id <i>required</i>	Example : UniqueId	UniqueId
name <i>required</i>	Example : Name	Name
order <i>required</i>	Example : Order	Order

SceneCollectionId

The id of the Scene Collection to which this member belongs.

Type : integer

SceneCollectionObject

Name	Description	Schema
------	-------------	--------

sceneCollection <i>required</i>	Example : SceneCollection	SceneCollection
-------------------------------------------	--------------------------------------------------	---------------------------------

SceneCollectionsResponse

Name	Description	Schema
sceneCollectionData <i>required</i>	Scene collection data for included scene collections. Example : [" SceneCollection "]	< SceneCollection > array
sceneCollectionIds <i>required</i>	Unique ids of all scene collections. Example : [" UniqueId "]	< UniqueId > array

ScheduledEvent

Name	Description	Schema
enabled <i>required</i>	Example : true	boolean
sceneId <i>optional</i>	Example : UniqueId	UniqueId
sceneCollectionId <i>optional</i>	Example : UniqueId	UniqueId
daySunday <i>required</i>	Example : true	boolean
dayMonday <i>required</i>	Example : true	boolean
dayTuesday <i>required</i>	Example : true	boolean
dayWednesday <i>required</i>	Example : true	boolean
dayThursday <i>required</i>	Example : true	boolean

dayFriday <i>required</i>	Example : true	boolean
daySaturday <i>required</i>	Example : true	boolean
eventType <i>required</i>	Example: EventType	EventType
hour <i>required</i>	Example : 0	integer
minute <i>required</i>	Example : 15	integer
id <i>required</i>	Example : UniqueId	UniqueId

ScheduledEventObject

Name	Description	Schema
scheduledEvent <i>required</i>	Example : ScheduledEvent	ScheduledEvent

ScheduledEventsResponse

Name	Description	Schema
scheduledEventData <i>required</i>	Scheduled event data for included scheduled events. Example : ["ScheduledEvent"]	< ScheduledEvent > array
scheduledEventIds <i>required</i>	Unique ids of all scheduled events. Example : ["UniqueId"]	< UniqueId > array

UniqueId

Unique resource identifier.

Type : integer