

## DALI-2 IoT

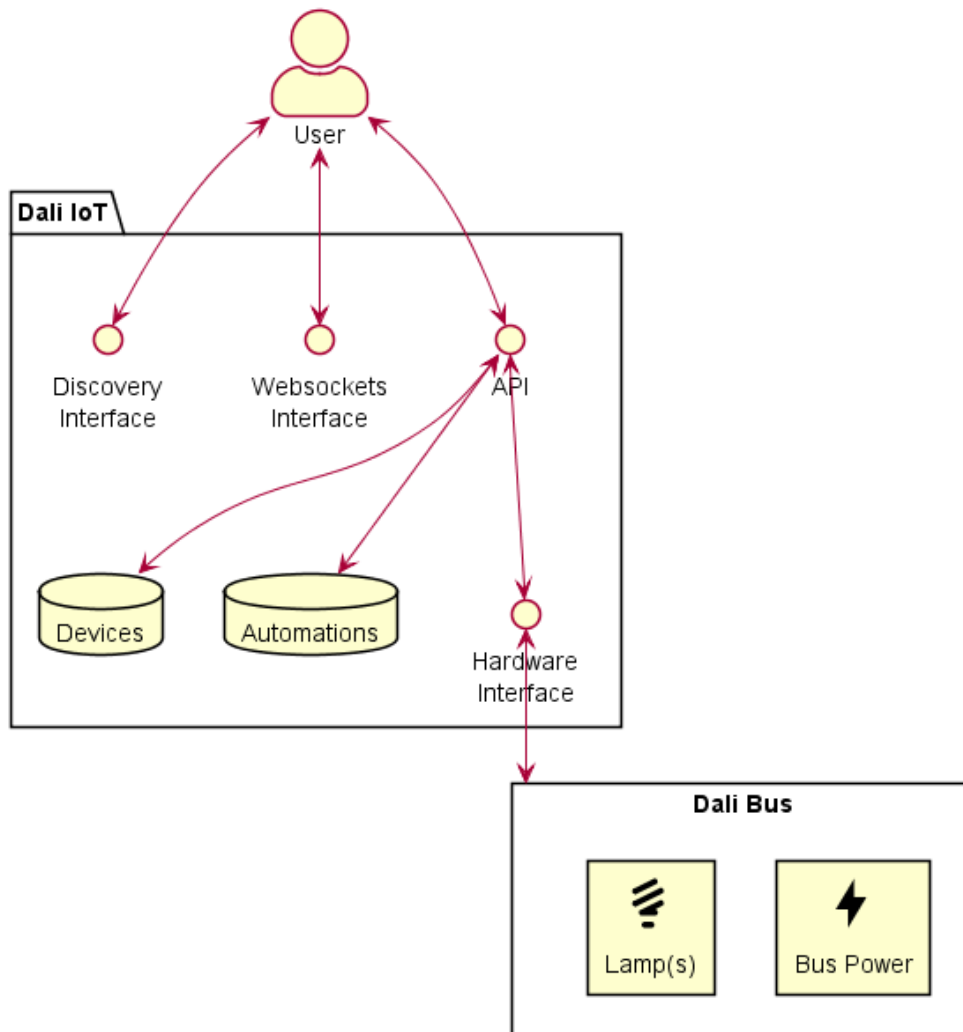
### Anleitung

#### API Dokumentation



Programmierschnittstellenbeschreibung  
des DALI-IoT Schnittstellen Moduls

Art.Nr. 89453886  
GTIN: 9010342013607



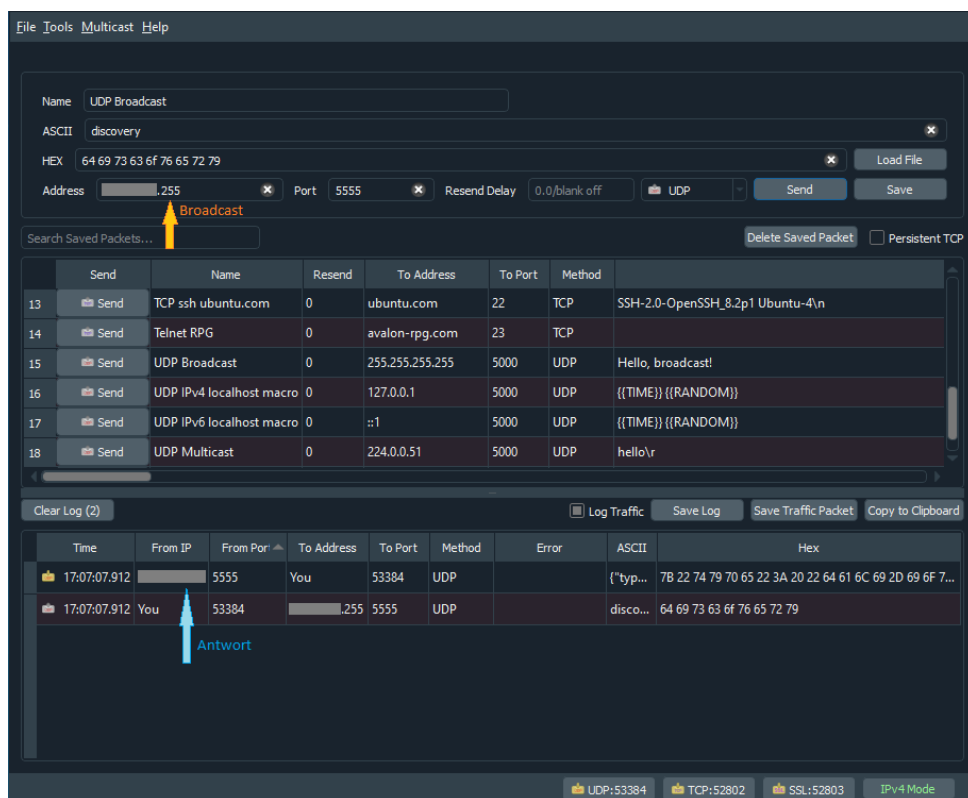
# Inhaltsverzeichnis

<b>1</b>	<b>Bestimmung der IP Adresse</b>	<b>4</b>
<b>2</b>	<b>Aufruf der DALI IoT API Dokumentation</b>	<b>5</b>
2.1	Kategorien und Endpunkte . . . . .	5
2.2	Liste der Schemata . . . . .	8
2.3	Direktes Testen der Endpunkte über die API Dokumentation . . . . .	9
<b>3</b>	<b>Ansteuern von DALI Geräten</b>	<b>11</b>
3.1	Das ControlData Objekt . . . . .	11
3.1.1	Ein- oder Ausschalten der Geräte . . . . .	11
3.1.2	Dimmen der Geräte . . . . .	11
3.1.3	Letztes aktives Level abrufen . . . . .	11
3.1.4	Aufrufen einer Szene . . . . .	11
3.1.5	Eine Szene abspeichern . . . . .	11
3.1.6	Konfigurieren von Farben mittels Rot-, Grün- und Blau (RGB) Werten . . . . .	12
3.1.7	Konfigurieren von Farbtemperatur . . . . .	12
3.1.8	Konfigurieren von Farben mittels Farbkoordinaten (XY) . . . . .	12
3.2	Ansteuern aller DALI Geräte auf dem DALI Bus . . . . .	12
3.3	Ansteuern einzelner Geräte . . . . .	13
3.3.1	Geräte Scan . . . . .	13
3.3.2	Abfrage der Geräte . . . . .	14
3.3.3	Befehle an Einzelgeräte senden . . . . .	14
3.4	Konfigurieren und Ansteuern von Gruppen . . . . .	15
3.4.1	Zuweisen von Gruppen . . . . .	15
3.4.2	Ansteuern von Gruppen . . . . .	16

# 1 Bestimmung der IP Adresse

Für die Nutzung der API Dokumentation ist es nötig die IP Adresse des DALI IoT Geräts zu kennen. Die IP Adresse kann anhand eines „Discovery“ Protokolls, welches von DALI IoT zur Verfügung gestellt wird, bestimmt werden. Das DALI IoT führt einen Service aus, der am Port 5555 auf UDP-Pakete mit dem Datenpaket `discovery` reagiert, indem es ein Datenpaket mit `{"type": "dali-iot"}` zurück sendet. Auf diese Weise lassen sich laufenden DALI IoT Instanzen in einem Netzwerk identifizieren, indem entweder ein UDP Broadcast-Paket oder einzelne UDP-Pakete an die jeweiligen Adressen im Netzwerk verschickt werden.

Ein simples Programm zum Versenden von Broadcasts ist „Packet Sender“. Um mit Packet Sender die DALI IoT Instanzen in einem Netzwerk zu finden, muss das „ASCII“-Eingabefeld auf `discovery` gesetzt werden, das „Address“-Feld auf die Broadcast-Adresse des Netzwerks (endet auf `.255`) bzw. eine einzelne Adresse die abgefragt werden soll, und das „Port“-Feld auf `5555`.



Klickt man nach diesen Einstellungen auf „Send“, dann wird ein UDP-Paket im Netzwerk versendet. Antworten auf dieses Paket scheinen im „Log“ unterhalb der Vorlagen für Pakete auf. Dort können die IP Adressen von DALI IoT Instanzen abgelesen werden.

## 2 Aufruf der DALI IoT API Dokumentation

Die API Dokumentation („docs“-Seite) des DALI IoT kann über `http://<IP_ADDRESS>/docs` mit einem Webbrowser aufgerufen werden, wobei `<IP_ADDRESS>` durch die IP Adresse des Geräts ersetzt werden muss. Auf dieser Seite sind zunächst die API Endpunkte, geordnet nach Kategorien (`control`, `devices`, `dali`, etc.) aufgelistet. In den einzelnen Kategorien befinden sich sogenannte Endpunkte mit denen mit dem DALI IoT System interagiert werden kann. Endpunkte sind eine Kombination aus Anforderungsmethode (GET, POST, PUT oder DELETE) und dem Endpunktnamen (z. B. `/broadcast/control`). Der gleiche Endpunktnamen kann für mehrere Endpunkte mit unterschiedlichen Anforderungsmethoden und Funktionalitäten genutzt werden.

**Dali IoT** 0.5.1 OAS3  
/openapi.json

This API documentation is **work in progress**. In addition to the REST API, there is also a **websocket API** which provides asynchronous events for status updates etc. Documentation for this websocket API will be available soon.

Endpoints to control devices

**control** ^

- **switchable**: Turns the device on or off
- **dimmable**: Dims the device to a given percentage
- **gotoLastActive**: Turns the device on to it's last active level
- **scene**: Set the device to a given scene
- **saveToScene**: Store the current configuration to scene
- **colorRGB**: Set the device to a given RGB value from 0.0-1.0
- **colorKelvin**: Set the device to a given color temperature in Kelvin
- **colorXY**: Set the device to a given XY color coordinate

POST	/device/{_id}/control	Control Device	∨
POST	/group/{_id}/control	Control Group	∨
POST	/broadcast/control	Control Broadcast	∨

**devices** Endpoints for information about devices ^

GET	/devices	Get Devices	∨
DELETE	/devices	Delete Devices	∨

### 2.1 Kategorien und Endpunkte

Beim Aufruf der API Dokumentation sind die Endpunktbeschreibungen zunächst zusammengeklappt; diese können mit einem Klick auf den Endpunkt aufgeklappt werden um weitere Details zu erhalten. Dabei werden eine kurze Beschreibung des Endpunkts, die zu übertragenden Parameter und die Rückmeldungen der API genauer ausgeführt. Als Beispiel ist hier der Endpunkt `GET /info` aufgeführt.

Eine Anforderung dieses Endpunkts benötigt keine Eingaben und überträgt als Rückgabewert von der API ein `InfoModel` Objekt. Ein Beispiel für die Struktur dieses Modells wird in der Dokumentation des Endpunkts gezeigt. Bei `GET /info` werden unter anderem der Name des Geräts und dessen Software- und Firmware-Versionen übertragen. Um ein genaueres Bild über die Zusammensetzung des Rückgabewerts zu erhalten lässt sich das Schema des Modells darstellen, indem man auf die Schaltfläche „Schema“ oberhalb des Beispiels klickt. Dabei werden die Datenfelder des Modells, deren Datentypen und eine kurze Erklärung der Daten dokumentiert.

Manche Endpunkte benötigen zusätzliche Angaben um eine Anforderung verarbeiten zu können. Zum

Beispiel muss für den Endpunkt `POST /device/{_id}/control`, mit dem einzelne Geräte angesteuert werden können, eine Identifikationsnummer und ein `ControlData` Objekt gesendet werden. Die Identifikationsnummer bestimmt welches Gerät angesteuert wird, während `ControlData` den gewünschten Zustand für das Gerät angibt. Ein Beispielwert für `ControlData` Objekte befindet sich in der Dokumentation des Endpunkts (Example Value).

Der Endpunkt `POST /device/{_id}/control`, enthält die Identifikationsnummer als Teil der Adresse des Endpunkts, während das `ControlData` Objekt in den Anfragedaten übermittelt wird. In der API Dokumentation wird die Identifikationsnummer durch ein separates Eingabefeld angegeben. Diese Nummer verändert den Endpunktnamen der Anfrage, was durch geschwungene Klammern im Endpunktnamen angezeigt wird. Um zum Beispiel ein Gerät mit Identifikationsnummer "id": 1 anzusteuern werden Anfragen an den Endpunkt `POST /device/1/control` gesendet.

**POST** /device/{\_id}/control Control Device

Control the features of one device by id

**Parameters** Try it out

Name	Description
<b>_id</b> * required integer (path)	<input type="text" value="_id"/>

**Request body** required application/json

Example Value | Schema

```
{
  "switchable": true,
  "dimmable": 50,
  "gotoLastActive": true,
  "emergency": true,
  "scene": 15,
  "saveToScene": 15,
  "colorRGB": {
    "r": 0,
    "g": 0.5,
    "b": 1
  },
  "colorKelvin": 4000,
  "colorXY": {
    "x": 0.432,
    "y": 0.15
  }
}
```

**Responses**

Das Schema dieses Datentyps lässt sich, so wie bei den Rückgabewerten, anzeigen, indem man auf „Schema“ oberhalb des Beispielwerts klickt, neben dem Button „Example Value“.

Example Value | Schema

```

ControlData ▾ {
  switchable          boolean
                      title: Switchable
                      example: true
  dimmable            number
                      title: Dimmable
                      example: 50
  gotoLastActive      boolean
                      title: Gotolastactive
                      example: true
  emergency            boolean
                      title: Emergency
                      example: true
  scene               integer
                      title: Scene
                      example: 15
  saveToScene         integer
                      title: Savetoscene
                      example: 15
  colorRGB             Colorrgb > {...}
                      example: OrderedMap { "r": 0, "g": 0.5, "b": 1 }
  colorKelvin          number
                      title: Colorkelvin
                      example: 4000
  colorXY             Colorxy > {...}
                      example: OrderedMap { "x": 0.432, "y": 0.15 }
}
    
```

## 2.2 Liste der Schemata

Anschließend an die Kategorien und Endpunkte befindet sich eine Liste von Schemata, die von den Endpunkten übertragen werden.



Diese können durch einen Klick auf das Schema ausgeklappt werden um deren Details anzuzeigen.



```
InfoModel {
  description:
    Model for querying general information about the device.

  name
    string
    title: Name
    default: dali-iot
    The user definable device name.

  version
    string
    title: Version
    default: vU.V.W/X.Y.Z
    The version string consisting of application version/firmware version.

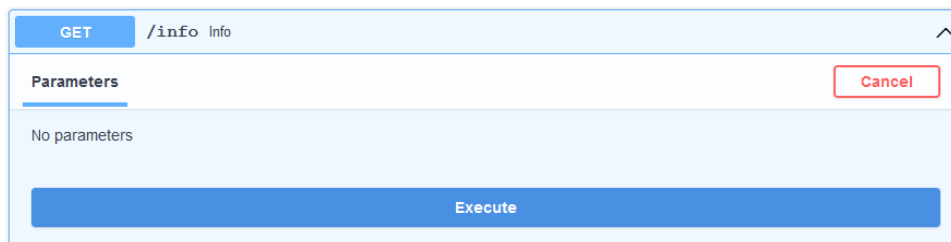
  tier
    string
    title: Tier
    default: basic
    The tier of the device € [basic, plus]. Plus devices support additional features, such
    as automations and addressing.

  errors
    Errors {
      description:
        A dictionary mapping error codes to their details.

      < * >:
        string
    }
}
```

## 2.3 Direktes Testen der Endpunkte über die API Dokumentation

Endpunkte können über den „Try it out“ Button direkt in der Dokumentation getestet werden. Durch eine Betätigung des Buttons öffnen sich zusätzliche Eingabefelder (die von der Beschaffenheit eines Endpunkts abhängen), sowie der Button „Execute“. Da beim Endpunkt `GET /info` keine Parameter vorhanden sind, ist in diesem Fall keine weitere Eingabe nötig.



Wird dieser Button betätigt, dann sendet der Browser eine Anforderung an den Endpunkt. Nach dem Absenden werden: ein Beispiel, wie eine äquivalenter Anforderung mit dem Programm „curl“ gesendet werden kann, die Adresse an welche die Anforderung geschickt wird, und die Antwort auf die Anforderung angezeigt. Darüber hinaus wird ein Button „Clear“ zum Entfernen der Anforderungen hinzugefügt.

Execute
Clear

**Responses**

**Curl**

```
curl -X 'GET' \
'http://192.168.0.62/info' \
-H 'accept: application/json'
```

**Request URL**

```
http://192.168.0.62/info
```

**Server response**

Code	Details
200	<p><b>Response body</b></p> <pre>{   "name": "dali-iot",   "version": "v0.5.1/0.0.184",   "tier": "basic",   "errors": {     "errorCode": "details"   } }</pre> <p style="text-align: right;"><span style="border: 1px solid #ccc; padding: 2px 5px;">Download</span></p> <p><b>Response headers</b></p> <pre>content-length: 94 content-type: application/json date: Wed,07 Jul 2021 14:39:56 GMT server: uvicorn</pre>

Da manche Endpunkte zusätzliche Angaben für deren Verarbeitung benötigen, können weitere Eingabefelder aufgeklappt werden. Zum Beispiel benötigt der Endpunkt `POST /device/{_id}/control` eine Identifikationsnummer um ein Gerät zu identifizieren, und ein `ControlData` Objekt, um dieses zu steuern. Pflichtfelder werden neben den Eingabefeldern mit „required“ gekennzeichnet. Felder, die keine Pflichtfelder sind, können leer belassen werden.

Cancel

**Parameters**

Name	Description
<p><b>_id</b> * required</p> <p>integer</p> <p>(path)</p>	<input style="width: 100%; border: 1px solid #ccc;" type="text" value="_id"/>

**Request body** required

application/json
▼

```
{
  "switchable": true,
  "dimmable": 50,
  "gotolastActive": true,
  "emergency": true,
  "scene": 15,
  "saveToScene": 15,
  "colorRGB": {
    "r": 0,
    "g": 0.5,
    "b": 1
  },
  "colorKelvin": 4000,
  "colorX": {
    "x": 0.432,
    "y": 0.15
  }
}
```

Execute

## 3 Ansteuern von DALI Geräten

### 3.1 Das ControlData Objekt

Geräte werden über ihre Features, wie z. B. `dimnable` für dimmbare Geräte, angesteuert. Um Geräte einzustellen, müssen deren Features und zugehörige Parameter durch ein `ControlData` Objekt angegeben werden. Das `ControlData` Schema ist eine Abbildung, die die gewünschten Werte den Namen der Features zuordnet. Einzelne Features sind optional; daher ist es ausreichend nur die gewünschten in einer Anforderung anzugeben. Es können mehrere, durch Beistriche getrennte Features gleichzeitig übertragen werden.

#### 3.1.1 Ein- oder Ausschalten der Geräte

Um Geräte ein- oder auszuschalten, wird das „switchable“-Feature genutzt, welches einen Boolean Wert benötigt, um das Gerät entweder ein (`true`) oder aus (`false`) zu schalten. Der „Request Body“ zum Einschalten aller Geräte setzt sich aus dem Namen des Features und dem Wert zusammen.

```
{
  "switchable": true
}
```

#### 3.1.2 Dimmen der Geräte

Um Geräte zu dimmen, wird das „dimmable“-Feature genutzt. Dieses akzeptiert eine Prozentzahl von 0 bis 100. Wird ein „dimmable“-Wert von 0 angegeben, dann werden die Geräte ausgeschaltet. Bei allen anderen Werten werden die Geräte auf das angegebene Level eingestellt.

```
{
  "dimmable": 50
}
```

#### 3.1.3 Letztes aktives Level abrufen

Um das letzte aktive Level wieder aufzurufen kann das „gotoLastActive“ Feature genutzt werden. Dabei wird ein `true`-Wert angegeben.

```
{
  "gotoLastActive": true
}
```

#### 3.1.4 Aufrufen einer Szene

Szenen können mit dem „scene“ Feature aufgerufen werden. Dabei wird die Nummer der Szene (0 bis 15) angegeben, welche aufgerufen werden soll.

Szenen werden in den jeweiligen DALI Geräten gespeichert. Viele DALI Geräte haben im Auslieferungszustand keine Szenenwerte vorgegeben. Zum Speichern von Szenenwerten, siehe Abschnitt 3.1.5.

```
{
  "scene": 15
}
```

#### 3.1.5 Eine Szene abspeichern

Um das aktuelle Level für eine Szene zu speichern kann das „saveToScene“ Feature genutzt werden. Dabei wird die Nummer der Szene (0 bis 15) angegeben, in die gespeichert werden soll.

```
{
  "saveToScene": 15
}
```

### 3.1.6 Konfigurieren von Farben mittels Rot-, Grün- und Blau (RGB) Werten

Um die Farbe einer Rot- Grün- und Blau- (RGB / RGBW) Leuchte einzustellen kann das „colorRGB“ Feature genutzt werden. Dabei wird eine Abbildung übertragen, die den Schlüsseln "r", "g" und "b" relative Farbwerte von 0 bis 1 zuordnet.

```
{
  "colorRGB": {
    "r": 0,
    "g": 0.5,
    "b": 1
  }
}
```

### 3.1.7 Konfigurieren von Farbtemperatur

Um die Farbe einer Farbtemperatur-Leuchte (CW-WW) einzustellen kann das „colorKelvin“ Feature genutzt werden. Dabei wird die Farbtemperatur in Kelvin angegeben (typischer Wertebereich von Lunatone CW-WW Dimmern: 100 K bis 20 000 K).

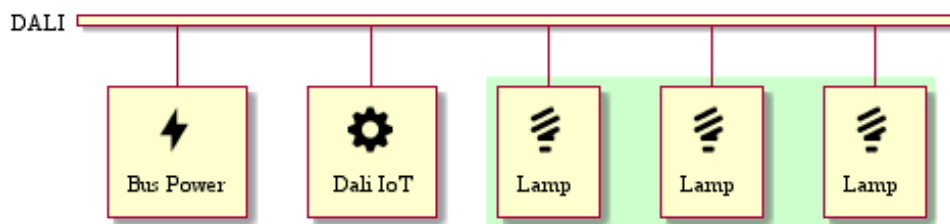
```
{
  "colorKelvin": 4000
}
```

### 3.1.8 Konfigurieren von Farben mittels Farbkoordinaten (XY)

Um die Farbe einer Leuchte mit Farbraumkoordinaten einzustellen kann das „colorXY“ Feature genutzt werden. Dabei wird eine Abbildung übertragen, die den Schlüsseln "x" und "y" die Farbkoordinaten von 0 bis 1 zuordnet.

```
{
  "colorXY": {
    "x": 0.432,
    "y": 0.15
  }
}
```

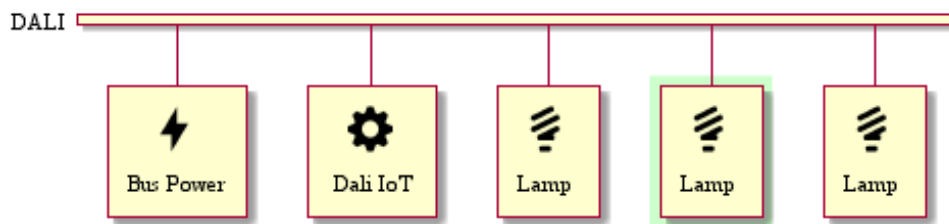
## 3.2 Ansteuern aller DALI Geräte auf dem DALI Bus



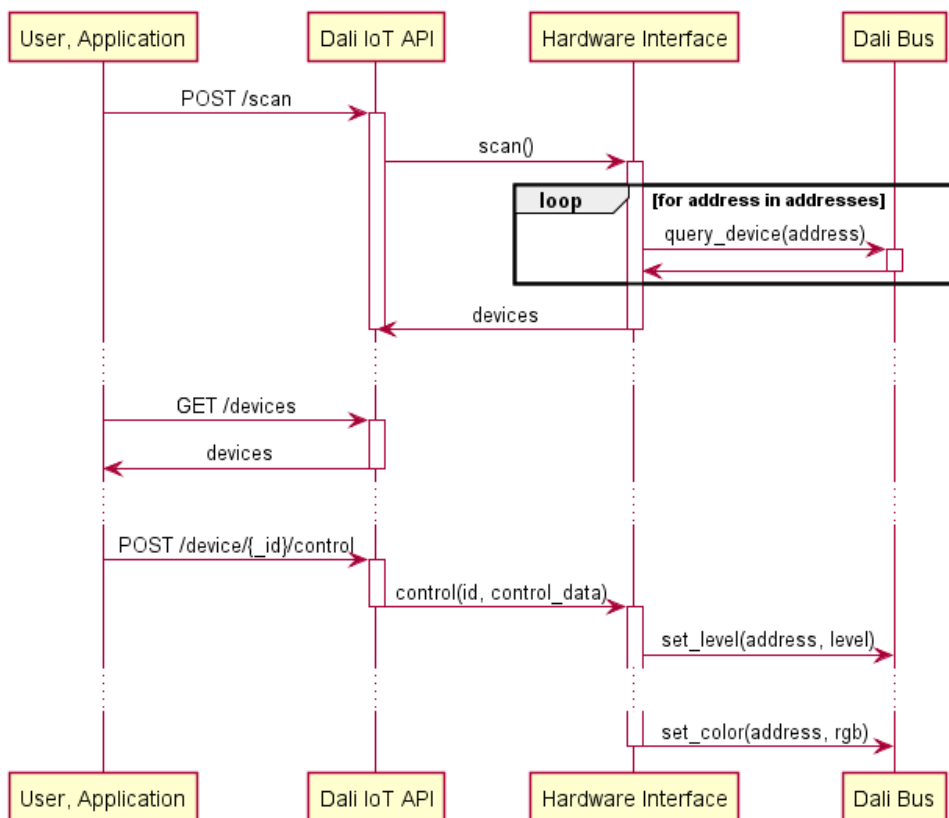
Um alle DALI-Geräte auf dem DALI Bus zu steuern ist keine zusätzliche Einrichtung notwendig; die Geräte können mit dem Endpunkt `POST /broadcast/control` direkt angesteuert werden. Dabei muss ein `ControlData` Objekt übertragen werden.

Zusätzlich kann ein optionaler Parameter `line` für die DALI Linie angegeben werden, zu welcher der Broadcast gesendet wird. In Normalfall (keine Angabe des Parameters) wird der Befehl an alle Linien gesendet. Unterstützt das Gerät z. B. 4 Linien, dann können Broadcasts unter Angabe des `line` Parameters auch ausschließlich an eine der Linien  $l \in [0, 1, 2, 3]$  gesendet werden.

### 3.3 Ansteuern einzelner Geräte



Um mit dem DALI IoT einzelne Geräte auf dem DALI Bus zu steuern, müssen diese zunächst einmalig erfasst und gespeichert werden. Jedes gespeicherte Gerät erhält dabei eine Identifikationsnummer, um es bei Anforderungen oder Befehlen von anderen Geräten zu unterscheiden. Nach dem Abschluss eines Geräte-Scans können die gespeicherten Geräte abgefragt werden, um deren Zustand und Identifikationsnummern zu erhalten. Mit diesen Nummern können dann einzelne Geräte angesteuert werden.



#### 3.3.1 Geräte Scan

Ein sogenannter Geräte-Scan dient dazu die Geräte auf dem DALI Bus zu adressieren, erfassen und speichern. Ein neuer Scan wird durch den `POST /dali/scan` Endpunkt gestartet, und benötigt zwei Angaben: `newInstallation` und `noAddressing`.

```
{
  "newInstallation": false,
  "noAddressing": false
}
```

`"newInstallation": true` wird bei einer Neuinstallation genutzt und bewirkt, dass die bekannten Geräte vor dem Scan gelöscht werden. Bei `"new Installation": false` werden die bestehenden Adres-

sen beibehalten und neu gefundene Geräte zu den bekannten hinzugefügt.

Außerdem kann vor der Erfassung der Geräte eine Adressierung gestartet werden, bei der den DALI-Geräten eine eindeutige Bus-Adresse zugewiesen wird. Falls dieser Schritt bereits anderweitig durchgeführt wurde (z. B. über eine zweite DALI IoT Instanz) müssen die DALI Geräte nur noch vom DALI IoT erfasst werden. In diesem Fall lässt sich die DALI Adressierung deaktivieren, indem der Wert `noAddressing` auf `true` gesetzt wird.

Da der Geräte-Scan alle Adressen des DALI Bus abfragt kann dieser bis etwa eine Minute dauern. Währenddessen kann der Fortschritt des Scans über den Endpunkt `GET /dali/scan` abgefragt werden. Dieser Endpunkt liefert als Antwort ein `ScanModel`, mit der Identifikationsnummer des aktiven Scans, dessen Fortschritt (in Prozent), der Anzahl gefundener Geräte und dem Status des Scans.

```
{
  "id": "e9160f03-0982-4cd7-88ab-00a4755bd17d",
  "progress": 38.671875,
  "found": 1,
  "status": "in progress"
}
```

Während ein Scan durchgeführt wird, kann kein neuer Scan gestartet werden. Allerdings können Scans abgebrochen werden, indem eine Anfrage an den Endpunkt `POST /dali/scan/cancel` geschickt wird.

### 3.3.2 Abfrage der Geräte

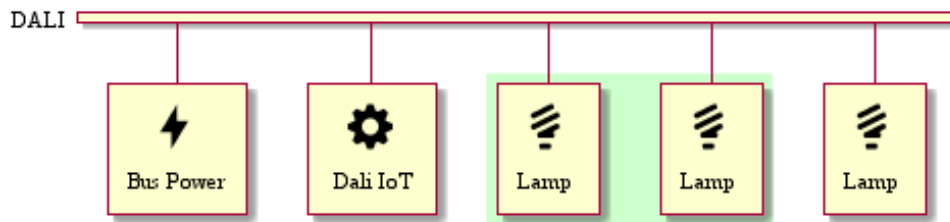
Die Geräte, die vom DALI IoT erfasst wurden, können mit dem `GET /devices` Endpunkt abgefragt werden. Dieser benötigt keine zusätzlichen Angaben und antwortet mit einer Liste der bekannten Geräte, deren Identifikationsnummern (`id`), Namen, Typen, den Zuständen ihrer Features, sowie deren Szenen- und Gruppenzugehörigkeit. Außerdem enthält die Antwort eine Signatur bestehend aus einem Zeitstempel und einem Zählstand.

```
{
  "devices": [
    {
      "id": 1,
      "name": "DALI #0",
      "info": "DALI #0",
      "type": "default",
      "features": {
        "switchable": {
          "status": false
        },
        "dimnable": {
          "status": 0
        },
        "scene": true,
        "colorRGB": {
          "status": {
            "r": 1,
            "g": 1,
            "b": 1
          }
        }
      },
      "colorKelvin": {
        "status": 2700
      },
      "colorXY": {
        "status": {
          "x": 0,
          "y": 0
        }
      },
      "saveToScene": true,
      "gotoLastActive": {}
    },
    "scenes": [],
    "groups": []
  ],
  { "id": 2, ... }, ...
},
  "timeSignature": {
    "timestamp": 1625747234.3620,
    "counter": 4
  }
}
```

### 3.3.3 Befehle an Einzelgeräte senden

Geräte können über den `POST /device/{_id}/control` Endpunkt einzeln angesteuert werden. Dabei wird, genauso wie beim Ansteuern aller Geräte, ein `ControlData` Objekt übermittelt. Zusätzlich zum `ControlData` Objekt in der „Request Body“ Eingabe muss allerdings die Identifikationsnummer des Geräts angegeben werden.

### 3.4 Konfigurieren und Ansteuern von Gruppen



Bevor Gruppen angesteuert werden können, müssen diese zunächst den DALI Geräten zugewiesen werden. Dieser Schritt kann übersprungen werden, falls die Geräte bereits Gruppen hinzugefügt wurden. Sind den Geräten Gruppen zugewiesen, dann können diese gemeinsam mit nur einer Anforderung angesteuert werden.

#### 3.4.1 Zuweisen von Gruppen

Um Gruppen zuzuweisen kann der Endpunkt `PUT /device/{_id}` genutzt werden. Dieser Endpunkt benötigt die Identifikationsnummer eines Geräts, sowie ein `DeviceUpdateModel`, bestehend aus einem (optionalen) Namen für das Gerät und einer (ebenso optionalen) Liste von Gruppennummern (0 bis 15). Um zum Beispiel die Gruppe 0 für ein Gerät einzustellen, muss ein „Request body“ mit dem Schlüssel `groups`, dem eine Liste mit der Nummer 0 zugewiesen ist, an den Endpunkt gesendet werden.

```
{
  "groups": [
    0
  ]
}
```

Der Rückgabewert der API enthält den Zustand des Geräts mit der bereits veränderten Gruppenzugehörigkeit. Diese Antwort ist ähnlich zu Antworten an Anforderungen des `GET /device/{_id}` Endpunkts – allerdings eingeschränkt auf das geänderte Gerät – und beinhaltet deshalb auch die Zustände der Features und die gespeicherten Szenen.

```
{
  "id": 1,
  "name": "DALI #0",
  "info": "DALI #0",
  "type": "default",
  "features": {
    "switchable": {
      "status": true
    },
    "dimnable": {
      "status": 100
    },
    "scene": true,
    "colorRGB": {
      "status": {
        "r": 1,
        "g": 1,
        "b": 1
      }
    }
  },
}
```

```
←
  "colorKelvin": {
    "status": 2700
  },
  "colorXY": {
    "status": {
      "x": 0,
      "y": 0
    }
  },
  "saveToScene": true,
  "gotoLastActive": {},
},
"scenes": [],
"groups": [
  0
],
"timeSignature": {
  "timestamp": 1625754320.503,
  "counter": 159
}
}
```

### 3.4.2 Ansteuern von Gruppen

Gruppen können über den `POST /group/{_id}/control` Endpunkt angesteuert werden. Dabei wird, genauso wie beim Ansteuern aller Geräte, ein `ControlData` Objekt übermittelt. Zusätzlich zum `ControlData` in der „Request Body“ Eingabe muss allerdings die Gruppennummer der Gruppe angegeben werden.