

Digitalstrom mit ioBroker steuern

Hier gehe ich darauf ein, wie man von ioBroker aus, Befehle an DigitalStrom sendet.

Später erkläre ich noch, wie man den aktuellen Digitalstrom-Status mit ioBroker abfragen kann.
Zunächst aber erstmal von ioBroker → DigitalStromServer

Vorraussetzungen ioBroker:

Scripts werden in Blockly geschrieben soweit möglich. Daher wird der JavaScript-Adapter benötigt.
Außerdem brauchen wir ein paar Objekte:

Objekte anlegen:

Oben auf das + und dort dann folgendes eingeben:

The screenshot shows the 'Neues Objekt hinzufügen' (Add new object) dialog in ioBroker. The title bar indicates the object being created is 'digitalstrom.0.Sessiontoken'. The form fields are as follows:

- Parent:** A greyed-out field.
- Name:** 'digitalstrom.0.Sessiontoken'
- Datenpunkt (Data Point):** A dropdown menu with 'Sessiontoken' selected.
- Typ (Type):** A dropdown menu with 'Zeichenkette' (String) selected.
- Datenpunkttyp (Data Point Type):** A dropdown menu with 'Zeichenkette' selected.

At the bottom right, there are two buttons: 'HINZUFÜGEN' (Add) and 'ABBRECHEN' (Cancel).

Ein „Punkt“ im Namen erzeugt immer einen Unterordner.

Mit dem Anlegen von digitalstrom.0.Sessiontoken habt ihr den Ordner digitalstrom.0 angelegt und auch gleich den ersten benötigten Datenpunkt „Sessiontoken“ (Zeichenkette)

Und zusätzlich noch einen Ordner pro Raum inkl. Datenpunkte anlegen:

The screenshot shows the 'Neues Objekt hinzufügen' (Add new object) dialog in ioBroker. The title bar indicates the object being created is 'digitalstrom.0.Wohnzimmer.LichtSceneState'. The form fields are as follows:

- Parent:** A greyed-out field.
- Name:** 'digitalstrom.0.Wohnzimmer.LichtSceneState'
- Datenpunkt (Data Point):** A dropdown menu with 'LichtSceneState' selected.
- Typ (Type):** A dropdown menu with 'Zahl' (Number) selected.
- Datenpunkttyp (Data Point Type):** A dropdown menu with 'Zahl' selected.

At the bottom right, there are two buttons: 'HINZUFÜGEN' (Add) and 'ABBRECHEN' (Cancel).

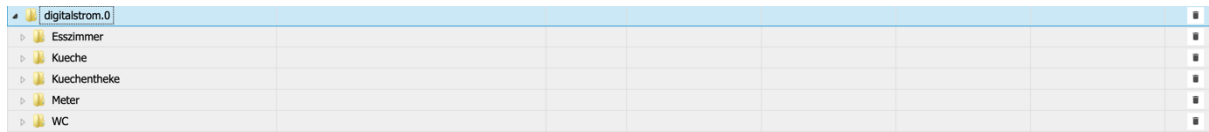
Hiermit legt ihr den Raum Wohnzimmer an und auch gleich den ersten benötigten Datenpunkt LichtSceneState (Zahl)

Jeder Raum enthält mindestens:

LichtSceneState (Zahl)

...erweiterbar um zB SchattenSceneState (Zahl), RaumSceneState (Zahl)

Beispiel für die entstandene Ordnerstruktur:



digitalstrom.0								
Esszimmer								
Kueche								
Kuechentheke								
Meter								
WC								

Vorraussetzungen DigitalStromServer:

Am Server per Browser anmelden (mit euren Zugangsdaten und ip-Adresse):

<https://dss.local:8080/json/system/login?user=dssadmin&password=dsspassword>

ApplicationToken erzeugen mit:

<https://dss.local:8080/json/system/requestApplicationToken?applicationName=ioBroker>

Den im Browser angezeigten Applicationtoken notieren/abspeichern!

Hintergrund:

Der Applicationtoken ist dauerhaft gültig und wird benötigt um einen Sessiontoken zu erzeugen, welcher dann benutzt wird, um den DS Server anzusteuern.

Scripten in ioBroker:

Jetzt geht es in Javascript / Blockly weiter:

Script 1: Sessiontoken

In ioBroker legen wir jetzt ein Script an, welches alle 3 Min den Sessiontoken erneuert und in den Datenpunkt „Sessiontoken“ schreibt. (Ein Sessiontoken ist bei Nichtnutzung max 3 min gültig, daher dieser Weg)



Funktionen: Javascript-Funktion



Trigger: Zeitplan

Funktionen: Token (entsteht nachdem ihr die Funktion oben angelegt und benannt habt)

Die hier gezeigte Javascriptfunktion „Token“ enthält folgendes Script:
(Rote Felder mit den Daten deines DigitalStromServers ersetzen)

```
var https = require("https");
var request = require("request");
var agentOptions;
var agent;

agentOptions = {
  host: '192.168.X.X'
, port: '8080'
, path: '/'
, rejectUnauthorized: false
};

agent = new https.Agent(agentOptions);




request({
  url:
  "https://192.168.X.X:8080/json/system/loginApplication?loginToken=DEINAPPTOKEN"
, method: 'GET'
, agent: agent
}, function (err, resp, body) {
  var response = JSON.stringify(body);
  var token = response.slice(26, -17);
  setState("digitalstrom.0.Sessiontoken"/*Sessiontoken*/, token);
  if (err) log(JSON.stringify(err));
  //if (body) log(JSON.stringify(body));
});
```

Abspeichern – fertig.

Jetzt solltet ihr alle 3 Minuten einen aktualisierten Sessiontoken erhalten!

Script 2: Szene an DigitalstromServer senden

In diesem Beispiel sende ich einen Befehl für meine Lichtszenen in der Küche.

	Funktionen: Javascript-Funktion
	Trigger: Falls Objekt (Hier den Datenpunkt der Lichtszene auswählen)
	Funktionen: Küche Szene(entsteht nachdem ihr die Funktion oben angelegt und benannt habt)

Die hier gezeigte Javascriptfunktion „Küche Szene“ enthält folgendes Script:

(Rote/blau Felder mit den Daten deines DigitalStromServers bzw deiner Datenpunkte ersetzen)

```
var https = require("https");
var request = require("request");
var agentOptions;
var agent;

agentOptions = {
  host: '192.168.X.X'
, port: '8080'
, path: '/'
, rejectUnauthorized: false
};

agent = new https.Agent(agentOptions);

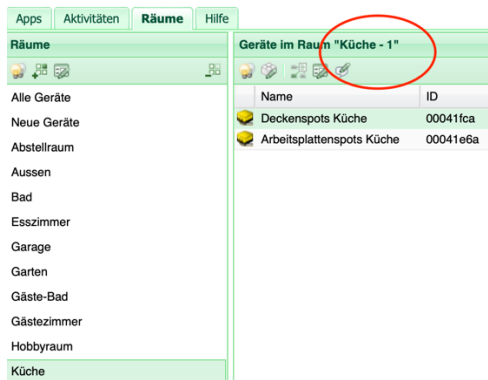
request({
  url:
    "https://192.168.X.X:8080/json/zone/callScene?&id=1&groupID=1&sceneNumber="+
    (getState("digitalstrom.0.Kueche.LichtSzeneState").val) + "&force=true&token="+
    String(getState("digitalstrom.0.Sessiontoken").val)
  , method: 'GET'
  , agent: agent
}, function (err, resp, body) {
  if (err) log(JSON.stringify(err));
});
```

Das blaue Feld (hier id=1) entspricht der DS Raum id die ihr ansteuern wollt.

In meinem Fall ist die Küche id=1

Raum IDs findet ihr auf der Oberfläche des DSS

digitalSTROM-Konfigurator



Die verwendete GroupID stellt immer das DS Gewerk dar. (0=Appartment wie zb Schlafen, Gehen, Kommen, 1=Licht, 2=Schatten, 3=Klima) Da es sich bei meinem Beispiel um eine Lichtszene handelt, ist die groupID=1
Infos dazu hier:

<http://developer.digitalstrom.org/Architecture/ds-basics.pdf>

getState müsst ihr noch mit dem passenden Datenpunkt befüllen! (Immer identisch mit der Trigger-Objekt-ID)

Abspeichern – fertig.

Für jeden Raum und jedes Gewerk legt ihr also so ein Script an.

Senden des eigentlichen Befehls.

Um Befehle zu senden müsst ihr jetzt wissen, welche Szene welcher Aktivität entspricht.
Das findet ihr hier:

<http://developer.digitalstrom.org/Architecture/ds-light.pdf>

<http://developer.digitalstrom.org/Architecture/ds-shade.pdf>

Lichtszene 1 am Schalter/Klemme entspricht der DSS-Szene 5
Lichtszene 0 am Schalter/Klemme entspricht der DSS-Szene 0
Lichtszene 2 am Schalter/Klemme entspricht der DSS-Szene 17
Usw (siehe Link)

Wenn ihr jetzt in den Datenpunkt die passende Szenennummer eingibt, sollte euer DS System darauf reagieren.
Empfohlen für den Test: 5 (Szene1) und 0 (Szene0/Aus)

Küche							
<input type="checkbox"/> ArbeitsplattenspotsDimmer	<input checked="" type="checkbox"/> KuecheArbeitsplattenspotsDimmer	state	state	Küche	Licht	0 %	<input type="checkbox"/>
<input type="checkbox"/> ArbeitsplattenspotsPower	<input checked="" type="checkbox"/> KuecheArbeitsplattenspotsPower	state	state	Küche		false	<input type="checkbox"/>
<input type="checkbox"/> DeckenspotsDimmer	<input checked="" type="checkbox"/> KuecheDeckenspotsDimmer	state	state	Küche	Licht	0 %	<input type="checkbox"/>
<input type="checkbox"/> DeckenspotsPower	<input checked="" type="checkbox"/> KuecheDeckenspotsPower	state	state	Küche		false	<input type="checkbox"/>
<input type="checkbox"/> LichtSzeneState	<input checked="" type="checkbox"/> KuecheLichtSzeneState	state	state	Küche		1	<input type="checkbox"/>

Hinweis:

Um zB „Gehen“ aufzurufen (eine globale, raumunabhängige Szene), braucht ihr einen Datenpunkt wie zb ApartementSzeneState

Im Script passt ihr id=0, groupID=0, an

Im Datenpunkt sendet ihr dann die Szenennummer=72