

<b>Ergänzende Information</b>			
Artikel-Nr.	X9750375.01	Auftrags-Nr.	
Produkt	Schnittstellenanleitung für colorSENSOR CFO100 CFO100(100) CFO200 CFO200(100)	Rahmen-Nr.	
Basis-Betriebsanleitung		colorSENSOR CFO – X9750375	

## Schnittstellenanleitung für den colorSENSOR CFO

# Inhalt

1	REST-API Kommunikation mit Controller .....	1
1.1	Basis-Information .....	1
1.1.1	API Endpoint .....	1
1.1.2	Beispiele für manuelle Abfrage .....	1
1.1.3	Abfrage-Beispiel .....	2
1.1.4	Schnellstart via API-Kommunikation .....	3
1.1.5	REST-API Einführung .....	4
1.1.5.1	Allgemeine Informationen .....	4
1.1.5.2	Response-Format und Fehlerbehandlung .....	5
1.1.5.3	Schaltausgänge, Trigger und Haltezeit .....	6
1.1.5.4	Websockets .....	7
1.1.5.5	Networking und Discovery .....	8
1.2	Resources .....	10
1.2.1	Controller Eigenschaften .....	10
1.2.1.1	Controller Eigenschaften abrufen .....	10
1.2.2	Sensor .....	11
1.2.2.1	Farbmesswerte auslesen .....	11
1.2.2.2	Letzte Farbmesswerte auslesen .....	13
1.2.2.3	Alle Farbgruppen (Matchers) auslesen .....	16
1.2.2.4	Erstelle eine neue Farbgruppe (Matcher) .....	18
1.2.2.5	Entferne mehrere oder alle Farbgruppen (Matchers) .....	22
1.2.2.6	Lese Farbgruppendetails (Matcher details) aus .....	22
1.2.2.7	Löschen einer Farbgruppe (Matcher) .....	24
1.2.2.8	Aktualisiere die Farbgruppenkonfiguration (Matcher Configuration) .....	25
1.2.2.9	Farben auslesen (Color Detectables) .....	29
1.2.2.10	Entferne mehrere oder alle Farben (ColorDetectables) .....	30
1.2.2.11	Erstelle Farben (ColorDetectables) .....	31
1.2.2.12	Farbe löschen (ColorDetectables) .....	33
1.2.2.13	Farbe ändern (ColorDetectable) .....	34
1.2.2.14	Farbe auslesen (ColorDecetables) .....	36
1.2.2.15	Wechsel aktuelles Erkennungsprofil .....	37
1.2.2.16	Erstelle ein Erkennungsprofil .....	38
1.2.2.17	Entferne mehrere oder alle Erkennungsprofile .....	47
1.2.2.18	Erkennungsprofile auslesen .....	48
1.2.2.19	Erkennungsprofil löschen .....	53
1.2.2.20	Erkennungsprofil ändern .....	53
1.2.2.21	Erkennungsprofil auslesen .....	63
1.2.2.22	Beginne Autogain-Prozedur .....	68

1.2.2.23	Anfrage benutzerdefinierte Weißreferenz .....	71
1.2.2.24	Festlegen einer benutzerdefinierten Weißreferenz .....	72
1.2.2.25	Trigger-Aktionen auslesen.....	72
1.2.2.26	Entferne mehrere oder alle Trigger-Aktionen .....	74
1.2.2.27	Erstelle Trigger-Aktionen .....	74
1.2.2.28	Lösche Trigger-Aktionen .....	75
1.2.2.29	Trigger-Aktionen ändern.....	76
1.2.2.30	Trigger-Aktionen auslesen.....	77
1.2.2.31	Controller Leistungsumfang auslesen .....	78
1.2.2.32	Farbräume auslesen .....	81
1.2.2.33	Farbraum abrufen .....	82
1.2.3	Einstellungen .....	83
1.2.3.1	Einstellungen exportieren .....	83
1.2.3.2	Einstellungen hochladen .....	83
1.2.3.3	Einstellungen importieren.....	84
1.2.3.4	Einstellungen zurücksetzen.....	85
1.2.4	System .....	85
1.2.4.1	Systemeinstellungen anfragen .....	85
1.2.4.2	Systemeinstellungen ändern .....	85
1.2.4.3	Zurücksetzen auf Werksfirmware oder –einstellungen .....	86
1.2.4.4	Neustart durchführen.....	87
1.2.4.5	Zeiteinstellungen auslesen .....	87
1.2.4.6	Zeiteinstellungen ändern .....	87
1.2.4.7	Unterstützte Zeitzonen abrufen .....	88
1.2.5	Netzwerk .....	89
1.2.5.1	Netzwerkeinstellungen zurücksetzen .....	89
1.2.5.2	Status und Konfiguration der Netzwerkschnittstellen auslesen .....	89
1.2.5.3	Status und Konfiguration einer einzelnen Netzwerkschnittstelle auslesen .....	91
1.2.5.4	IPv4 und/oder IPv6 Adresskonfiguration .....	92
1.2.6	Peripheriegeräte .....	96
1.2.6.1	Tastaturinformationen ausgeben.....	96
1.2.6.2	Tastatur modifizieren .....	97
1.2.6.3	Benutzerinteraktionen der Tastatur auslesen .....	98
1.2.6.4	Auflistung verfügbarer Eingabeelemente .....	99
1.2.6.5	Benutzerinteraktion auf der Tastatur simulieren .....	99
1.2.6.6	Ausgabekonfiguration abrufen .....	100
1.2.6.7	Ausgabekonfiguration anpassen.....	100
1.2.6.8	Aktuelle RS232 Schnittstellenkonfiguration abrufen .....	101
1.2.6.9	RS232 Schnittstellenkonfiguration anpassen.....	101
1.2.6.10	Triggerergebnis-Statistik abrufen.....	103

1.2.6.11	Aktuelle USB-Schnittstellenkonfiguration abrufen.....	103
1.2.6.12	USB-Schnittstellenkonfiguration anpassen .....	104
1.2.7	Aktionen.....	105
1.2.7.1	Aktionsbeschreibungen.....	105
1.2.7.2	Aktion „enable_switching_output“.....	105
1.2.7.3	Aktion „teach_single“.....	105
1.2.7.4	Aktion „keylock“.....	105
1.2.7.5	Aktion „run_autogain“.....	106
1.2.7.6	Aktion „remove_all_detectables“.....	106
1.2.7.7	Aktion „remove_all_matchers“.....	106
1.2.7.8	Aktionen auflisten .....	106
1.2.7.9	Aktion abrufen.....	106
1.2.7.10	Software-ausgelöste Aktionen .....	107
1.2.8	Standards .....	113
1.2.8.1	Standardwertepaare auflisten.....	113
1.2.8.2	Standardwertepaare erstellen .....	114
1.2.8.3	Mehrere oder alle Standardwertpaare entfernen .....	116
1.2.8.4	Standardwertepaare anpassen .....	116
1.2.8.5	Standardwertepaar löschen .....	117
1.2.8.6	Standardwertepaar abrufen.....	118
1.2.9	Firmware .....	118
1.2.9.1	Firmware-Informationen abrufen .....	118
1.2.9.2	Firmware-Image-Uploads .....	119
1.2.9.3	Firmware hochladen .....	120
1.2.9.4	Firmware-Upload-Methoden.....	120
1.2.9.5	Fehlerbehandlung .....	121
1.2.9.6	Firmware-Image-Datei abrufen .....	122
1.2.9.7	Firmware-Image-Datei löschen.....	123
1.2.9.8	Wechsel zu neuer Firmware .....	123
1.2.9.9	Datensatz der Firmware hochladen.....	124
1.2.9.10	Wiederherstellungs-Firmware-Informationen abrufen .....	124
1.2.9.11	Wiederherstellungs-Firmware aktualisieren .....	125
1.2.9.12	Lese Firmware-Einstellungen aus.....	125
1.2.9.13	/firmware/settings .....	126
1.2.9.14	Firmware-Status abrufen .....	127
1.2.10	Zugriffskontrolle .....	127
1.2.10.1	Benutzer.....	127
1.2.10.2	Rollen .....	127
1.2.10.3	Auswertung der Zugriffsberechtigungen.....	128
1.2.10.4	HTTP Antworten.....	128

1.2.10.5	Untersuche Bereiche der Zugriffskontrolle und Aktionen .....	128
1.2.10.6	In ein Konto einloggen.....	129
1.2.10.7	Informationen über die aktuell genutzten Anmeldedaten abrufen .....	131
1.2.10.8	Logout / Alle aktuellen Anmeldeinformationen ungültig machen .....	132
1.2.10.9	Benutzer anlegen.....	132
1.2.10.10	Entferne mehrere oder alle Benutzer.....	133
1.2.10.11	Benutzer auflisten .....	133
1.2.10.12	Benutzer löschen .....	134
1.2.10.13	Benutzer anpassen .....	134
1.2.10.14	Benutzer abrufen .....	135
1.2.10.15	Zugriffsrollen auflisten.....	136
1.3	Typenbezeichnung .....	138
1.3.1	AccessAction .....	138
1.3.2	Zugriffskontrollfunktionen .....	138
1.3.3	Zugriffsgenehmigung .....	138
1.3.4	Zugriffsrolle .....	138
1.3.5	Zugriffsbereich.....	139
1.3.6	Aktion .....	139
1.3.7	Aktion Aktivierte Schaltausgang .....	139
1.3.8	Aktion Tastatursperre.....	139
1.3.9	Aktion Entferne alle Farben (Detectables).....	140
1.3.10	Aktion Entferne alle Farbgruppen (Matchers) .....	140
1.3.11	Aktionsergebnis Aktiviere Schaltausgang .....	140
1.3.12	Aktionsergebnis Tastatursperre.....	143
1.3.13	Aktionsergebnis Entferne alle Farben (Detectables).....	143
1.3.14	Aktionsergebnis Entferne alle Farbgruppen (Matchers) .....	143
1.3.15	Aktionsergebnis Autogain ausführen .....	143
1.3.16	Aktionsergebnis Farbe (Detectable) einlernen .....	144
1.3.17	Aktion Autogain ausführen .....	145
1.3.18	Aktion Farbe (Detectable) einlernen .....	145
1.3.19	Aktionsauslöser .....	146
1.3.20	Alias .....	146
1.3.21	Amplifikationslevel .....	146
1.3.22	Jede Aktion .....	146
1.3.23	Jedes Aktionsergebnis .....	147
1.3.24	Autogain-Einstellungen.....	147
1.3.25	Durchschnittliche Stichprobenzahl.....	147
1.3.26	Grundfarbtoleranz.....	148
1.3.27	Serielle Grundeinstellungen .....	148
1.3.28	Box-Farbtoleranz .....	148

1.3.29	Farbwertkoordinate.....	148
1.3.30	Farbe (ColorDetectable) .....	148
1.3.31	Farberkennungsergebnis .....	149
1.3.32	Farberkennungsergebnisliste .....	151
1.3.33	Farberkennungsergebnis oder Null.....	151
1.3.34	Farbgruppe (ColorMatcher).....	152
1.3.35	Farbanpassungsergebnis.....	153
1.3.36	Farbdarstellung.....	154
1.3.37	Farbraum .....	155
1.3.38	Farbraumachsen.....	155
1.3.39	Farbraum-ID.....	156
1.3.40	Farbraumtoleranzkarte .....	156
1.3.41	Farbtoleranz.....	156
1.3.42	Kompensationseinstellungen .....	157
1.3.43	CorrectedColor .....	157
1.3.44	Aktuelle Erkennungsprofil-ID .....	158
1.3.45	Aktueller Schaltausgänge-Zustand .....	158
1.3.46	Zylinder-Farbtoleranz.....	158
1.3.47	Standardwertepaare .....	158
1.3.48	Erkennungsprofil .....	159
1.3.49	Geräteinformationen .....	163
1.3.50	Geräteseriennummer .....	164
1.3.51	Geräteherstellerschlüssel .....	164
1.3.52	Geräteherstellername .....	164
1.3.53	Fehler .....	164
1.3.54	Firmware-Build-ID .....	164
1.3.55	Firmware-Image-Datei .....	165
1.3.56	Firmware-Image Größe.....	165
1.3.57	Firmware-Image Upload .....	165
1.3.58	Firmware-Informationen.....	166
1.3.59	Firmware-Wiederherstellungsinformationen.....	166
1.3.60	Firmware-Ausführungsinformationen .....	166
1.3.61	Firmware-Einstellungen .....	167
1.3.62	Firmware Version .....	167
1.3.63	Hash Digest .....	167
1.3.64	Haltezeit .....	167
1.3.65	Hostname .....	168
1.3.66	Unbegrenzte Farbtoleranz.....	168
1.3.67	Zustand der Eingänge .....	168
1.3.68	Schnittstelle RS232.....	168

1.3.69	Schnittstelle USB .....	168
1.3.70	Tastatur-Ereignis.....	169
1.3.71	Tastatur-Ereignis-Eingabe .....	169
1.3.72	Tastatur-Ereignis-Name .....	169
1.3.73	Tastaturanzeige .....	169
1.3.74	Tastaturinformationen.....	170
1.3.75	Tastatureingabetaste .....	171
1.3.76	Anmeldeinformationen .....	171
1.3.77	MAC-Adresse.....	172
1.3.78	Netzwerk-Adresskonfiguration IPv4.....	172
1.3.79	Netzwerk-Adresskonfiguration IPv4DHCP.....	172
1.3.80	Netzwerk-Adresskonfiguration IPv4Static .....	172
1.3.81	Netzwerk-Adresskonfiguration IPv6.....	172
1.3.82	Netzwerk-Adresskonfiguration IPv6Auto .....	172
1.3.83	Netzwerk-Adresskonfiguration IPv6DHCP.....	172
1.3.84	Netzwerk-Adresskonfiguration IPv6Static .....	173
1.3.85	Netzwerkadresse IPv4 .....	173
1.3.86	Netzwerkadresse IPv6 .....	173
1.3.87	Informationen zur Konfiguration der Netzwerkschnittstellen-Adresse .....	173
1.3.88	Konfiguration der Netzwerkschnittstellen-Adresse IPv4.....	174
1.3.89	Konfiguration der Netzwerkschnittstellen-Adresse IPv6.....	174
1.3.90	Konfigurationszustand der Netzwerkschnittstellenadresse.....	174
1.3.91	Netzwerkschnittstellenadressen IPv4 .....	176
1.3.92	Netzwerkschnittstellenadressen IPv6 .....	176
1.3.93	Netzwerkschnittstellenadressen-Familieninformationen IPv4 .....	176
1.3.94	Netzwerkschnittstellenadressen-Familieninformationen IPv6 .....	177
1.3.95	Netzwerkschnittstellenadresse Familienzustand IPv4.....	178
1.3.96	Netzwerkschnittstellenadresse Familienzustand IPv6.....	178
1.3.97	Netzwerkschnittstellenadresse IPv4 .....	179
1.3.98	Netzwerkschnittstellenadresse IPv6 .....	179
1.3.99	Netzwerkschnittstelleninformationen.....	179
1.3.100	Netzwerkschnittstellenname.....	181
1.3.101	Statische Daten der Netzwerkschnittstelle .....	181
1.3.102	Normierungskonstante .....	181
1.3.103	Freigabekanal .....	182
1.3.104	Abtaste.....	182
1.3.105	Stichprobeneinstellungen.....	182
1.3.106	Sensor-Fähigkeiten.....	183
1.3.107	Serien-Eliza-Einstellungen .....	190
1.3.108	Serien-Modbus-Einstellungen .....	190

1.3.109	Signalfarbe.....	190
1.3.110	Kugel-Farbtoleranz .....	190
1.3.111	Unterstützte Zeitzonen.....	190
1.3.112	Schaltausgangstreiber.....	190
1.3.113	Schaltausgänge .....	191
1.3.114	Schaltausgänge beschreibbar .....	191
1.3.115	Systemeinstellungen .....	191
1.3.116	Systemzeiteinstellungen .....	191
1.3.117	Zeitstempel .....	192
1.3.118	Zeitstempel Backend-Uptime .....	192
1.3.119	Toleranzformname.....	192
1.3.120	Transformierte Farbe .....	192
1.3.121	Auslöseereignis .....	192
1.3.122	Triggerereignisname.....	192
1.3.123	Auslösequelle .....	193
1.3.124	Zustand der Auslösequelle .....	193
1.3.125	Benutzer.....	194
1.3.126	UUID .....	194
1.3.127	Gewünschter Schaltausgangszustand.....	194
1.3.128	Weißreferenz .....	195
1.3.129	Netzwerkschnittstellenadresse IPv4 .....	195
1.3.130	Netzwerkschnittstellenadresse IPv6 .....	195
2	RS232 bzw. USB.....	196
2.1	Vorwort.....	196
2.1.1	Verbindungsdetails .....	196
2.1.2	Syntax .....	196
2.1.3	Ausgabeformat & Nachrichten-Parsing .....	197
2.1.4	Nomenklatur .....	197
2.1.5	Typische Muster .....	197
2.1.6	Unterschiede im Vergleich zu der REST-API.....	198
2.2	Befehlsreferenz .....	199
2.2.1	Zugriffsbefehl .....	199
2.2.2	Gerätebefehl .....	199
2.2.3	Firmware-Befehl.....	199
2.2.4	Hilfsbefehl .....	199
2.2.5	Tastaturbefehl .....	200
2.2.6	Farbgruppenbefehl .....	200
2.2.7	Netzwerkbefehl .....	202
2.2.8	Wiederholungsbefehl.....	202
2.2.9	Stichprobenbefehl .....	203

2.2.10	Sensorbefehl.....	203
2.2.11	SET-Befehl .....	204
2.2.12	Systembefehl .....	204
3	Modbus Dokumentation .....	205
3.1	Einführung .....	205
3.2	Schnellstart .....	206
3.3	Unterstützte colorSENSOR-Funktionen .....	207
3.4	Datentypen und Registeradressierung.....	208
3.4.1	Datentypen und Modbus-Funktionen.....	208
3.4.2	Register Adressen .....	208
3.4.3	Einfache Datentypen .....	208
3.5	Sitzungsstatus, Aktualität und mehrere Schnittstellen .....	210
3.5.1	Funktionen .....	210
3.5.1.1	API-Endpoint „Einstellen und aktivieren der automatischen Aussteuerung“: /sensor/detection-profiles/current/autogain.....	210
3.5.1.2	API-Endpoint „Weiß-Referenz setzen / zurücksetzen“: /Sensor/Erkennungs- Profile/Strom/Weiß-Referenz .....	210
3.5.1.3	API-Endpoint „Neue Farbtabelle hinzufügen“: /sensor/matchers .....	210
3.5.1.4	API-Endpoint „Verwalten der Farben einer Farbgruppe“: /sensor/matchers .....	211
3.5.1.5	API-Endpoint "Sensorfähigkeiten lesen": /sensor/capabilities .....	211
3.5.1.6	API-Endpoint „Aktuelle Messwerte abrufen“: /sensor/samples/current .....	211
3.5.1.7	API-Endpoint „Status der Farbtabellen“: /sensor/detection-profiles/current 212	
3.5.1.8	API-Endpoint „Farbtabellen löschen“: /sensor/matcher .....	212
3.5.1.9	API-Endpoint „Schaltausgangstreiber einstellen“: /peripherals/outputs .....	212
3.5.1.10	API-Endpoint „Firmware-Version abfragen“: /firmware .....	212
3.5.1.11	API-Endpoint „Einstellungen zurücksetzen“: /settings.....	213
3.5.1.12	API-Endpoint „Auf Werkseinstellung zurückstellen“: /system-/factory-reset 213	
3.5.1.13	API-Endpoint „System Neustart“: /system/reboot.....	213
3.5.1.14	API-Endpoint „Upgrade auf Wiederherstellungs-Firmware“: /system-/factory- reset 213	
3.5.1.15	API-Endpoint „RS232-Schnittstelle konfigurieren“: /peripherals/rs232.....	213
3.5.1.16	API-Endpoint „USB-Schnittstelle konfigurieren“: /peripherals/usb.....	214
3.5.1.17	Adressen zum Testen der Datenformate .....	214

## 1 REST-API Kommunikation mit Controller

### 1.1 Basis-Information

Die HTTP-basierte API<sup>1</sup> ermöglicht die Abfrage und Änderung der Controller-Konfiguration bzw. aktueller Zustände.

Es ist die primäre Schnittstelle, über diese sind alle Details des Controllers zugänglich. Beispielsweise kommuniziert auch die integrierte Web-Anwendung, RS232, USB, MEDAQUlib und Modbus ausschließlich über die API mit dem Controller.

Die HTTP-API ermöglicht somit die Integration der Controller in eigene Werkzeugketten und Prozesse (Datensammlung, ERP-Einbindung, Backup, Monitoring, ...).

Die API ist REST-artig und unterstützt somit folgende HTTP-Requests, um Aktionen mit unterschiedlichen Auswirkungen zu unterscheiden:

- **GET**
  - Status einer Ressource (z.B. ein aktuelles Sample); **GET**-Abfragen sind frei von Nebeneffekten und werden manchmal im Cache-Speicher abgelegt.
- **POST**
  - Erzeugen einer neuen Ressource (z.B. eine neue Farbe oder Farbgruppe einlernen)
- **PUT**
  - Änderung einer Ressource (z.B. Änderung einer Toleranz in einer Farbgruppe / Matcher)
- **DELETE**
  - Löschen, Zurücksetzen oder Leeren einer Ressource (z.B. Löschen einer Farbe oder zurücksetzen auf Werkseinstellung)

1) Version 1.5.10

#### 1.1.1 API Endpoint

Die API-Endpunkte sind hier als Zusatz zur Basis URL (`http://`) zu verstehen. Als Beispiel sei hier der Browseraufruf (**GET**-Request) für den colorSENSOR CFO zur Abfrage der Ausgangskonfiguration genannt:

`http://169.254.168.150/api/peripherals/outputs`

Basis-URL | API-Endpoint

`http://{sensor}/api`

<b>sensor</b> String, required	Hostname oder IP-Adresse des Controllers
-----------------------------------	--

#### 1.1.2 Beispiele für manuelle Abfrage

Die folgenden manuellen Abfragen sollen lediglich die beispielhafte Nutzung der API veranschaulichen.

Für die Einbindung in eigene Werkzeuge stellen die meisten Programmiersprachen und Entwicklungsumgebungen typischerweise HTTP-Programmierschnittstellen für den einfachen Zugriff zur Verfügung.

Die folgenden empfohlenen Tools zur API-Abfrage sind nicht zwingend erforderlich, können aber verwendet werden.

Umgebung	Werkzeug-Empfehlung
Kommandozeile (CLI)	cURL
Windows PowerShell	Invoke-RestMethod
Chrome/Chromium	Postman

### 1.1.3 Abfrage-Beispiel

Jeder Web-Browser nutzt **GET** Requests zum Abrufen von Inhalten. Mit Werkzeugen ist es einfach, Anfragen auch mit anderen HTTP-Verben zu senden. In Anwendung auf den colorSENSOR CFO ergibt zum Beispiel die Anfrage

```
curl -X GET http://169.254.168.150/api/sensor/samples/current
```

folgende Antwort in Form eines JSON-Pakets:

```
{
  "data": {
    "timestamp": 3145601368.0,
    "uuid": "0f721d83-f3c0-4584-8913-a51a5b842784",
    "transformed_color": {
      "values": [ 99.953887939453125, -0.0064074993133544922,
0.017380714416503906 ]
    },
    "corrected_color": {
      "values": [ 0.79777300357818604, 0.74252212047576904, 0.28755432367324829
]
    },
    "detection": {
      "matcher": null,
      "output_pattern": { "states": [ false, false, false ] },
      "distances": [ null, null, null ]
    },
    "representations": {
      "RGB": [ 0.9994870320649919, 0.99951960105113213, 0.99927028464270895 ]
    },
    "inputs": {
      "trigger_0_down": true,
      "trigger_1_down": true,
      "trigger_2_up": false,
      "trigger_0_up": false,
      "trigger_3_down": true,

```

```
"trigger_3_up": false,  
"trigger_2_down": true,  
"trigger_1_up": false  
}  
,  
"errors": []  
}
```

#### 1.1.4 Schnellstart via API-Kommunikation

Die folgende Abfolge von Requests konfiguriert den Controller für die Erkennung einer gewünschten Farbe.

Die folgenden Abfragebeispiele werden über curl Befehle (Beispiele mit Präfix \$) oder Windows-Powershell (mit Präfix >) gesendet.

Dieselben requests können auch in allen anderen Programmiersprachen verwendet werden, die HTTP Abfragen unterstützen.

1. Löschen Sie alle Einstellungen:

```
$ curl -X DELETE http://sensor/api/settings
```

```
> Invoke-RestMethod -Method DELETE -Uri http://<sensorIP>/api/settings
```

2. Platzieren Sie ein neutrals weißes Farbobjekt vor die Optik des Sensors.
3. Passen Sie die Sensoreinstellungen an das optische Setup an, indem Sie den *Autogain*-Vorgang einleiten:

```
$ curl -X POST http://<sensorIP>/api/sensor/detection-profiles/current/autogain
```

```
> Invoke-RestMethod -Method POST -Uri http://<sensorIP>/api/sensor/detection-profiles/current/autogain
```

4. Platzieren Sie das Zielobjekt vor der Optik des Sensors.

5. Lernen Sie dessen Farbe ein:

```
$ curl -X POST http://<sensorIP>/api/sensor/detectables
```

```
> Invoke-RestMethod -Method POST -Uri http://<sensorIP>/api/sensor/detectables
```

6. Der Sensor sollte nun seinen ersten Ausgang (OUT0) auf High setzen, solange sich das Zielobjekt davor befindet.

- Der erste Schaltausgang ist standardmäßig der ersten Farbe zugeordnet. Dies ist konfigurierbar.

7. Fordern Sie die aktuellen „current sample“ Daten an und wählen Sie nur die Farbwerte aus:

```
# hint: "jq" is a separate tool for querying JSON datasets
```

```
$ curl http://<sensorIP>/api/sensor/samples/current | jq .data.trans-  
formed_color.values
```

- Ergebnis:

```
[ 101.23353576660156, 7.889449596405029, -42.49897003173828 ]
```

### 1.1.5 REST-API Einführung

#### 1.1.5.1 Allgemeine Informationen

##### Collections (Arrays bzw. Listen)

- können via **GET** Request abgefragt werden.
- Einige Abfragen können über Parameter gefiltert werden, die unter RESOURCES A 3.2 aufgeführt sind.
- Einige Collections erlauben **POST** Anfragen für die Erzeugung zusätzlicher Elemente der Collection.
  - Eigenschaften neuer Elemente können normalerweise im **POST** Request festgelegt werden.
  - Ein Fehler beim Erstellen eines neuen Elements aufgrund fehlerhafter Daten wird mit einem 400 HTTP Statuscode angezeigt.
  - Ein Fehler beim Erstellen eines neuen Elements, der durch zu viele Elemente in der Collection verursacht wurde, wird durch einen 422 HTTP Statuscode angezeigt.
  - Einige Collections können geleert werden, indem eine **DELETE** Befehl an sie gesendet wird, der dieselben Abfrageparameter wie der **GET** Request unterstützt, sofern vorhanden.
  - Eine **DELETE** Request für die Collection gibt immer einen positiven HTTP Statuscode wie 200 oder 204 zurück, ohne Rücksicht auf die Anzahl der gelöschten Elemente (auch wenn Aufgrund des Befehls keine Elemente gelöscht wurden).

##### Collection Items

- Auf einzelne collection items kann normalerweise über einen URL-Pfad wie `/collection-path/item-id` zugegriffen werden (z. B. `/api/access/users/barbara`, wobei `/users` die collection und `barbara` die Element-ID ist).
  - Wenn kein Element mit dieser spezifischen ID vorhanden ist, entweder weil es in der Zwischenzeit als Antwort auf eine Benutzer Request gelöscht wurde oder (bei größen- oder zeitlich begrenzten Collections) das Element entfernt wurde, wird die Anforderung mit einem HTTP Statuscode 404 beendet.
  - Die Daten einiger Collections Items können mit **PUT** Request aktualisiert werden, wobei der Body ein JSON Objekt mit den Daten enthält, die aktualisiert werden sollen. Teilaktualisierungen werden ebenfalls unterstützt. Das Ändern von schreibgeschützten Attributen kann mit einem HTTP Statuscode 400 beendet werden. Die Antwort für eine erfolgreiche Aktualisierung enthält den Status der Ressource nach der Änderung.
  - Die eindeutige ID eines Elements (meistens ein `uuid`) ist unveränderlich.
  - Einige Collection Items können mit einer **DELETE** Request entfernt werden.
- Einige Collections unterstützen mehr als ein URL-kompatibles Element ID Feld. Die `matcher` (Farbgruppe), `detectable` (Farbtabelle) und `detection profile` (Erkennungsprofil) Collection Items können beispielsweise auch über ihr jeweiliges `alias` Feld abgerufen werden.

##### Datenformatierungskonvention

- Die Bezeichnungen sind in amerikanischem Englisch
- Zusammengesetzte Wörter in Resource URLs werden getrennt (z. B. `white-reference`).

- Zusammengesetzte Wörter in JSON Objekten oder Daten, die als mehrteilige / Formulare Daten akzeptiert werden, verwenden die `>>snake_cased<<` Schreibweise (z. B. `white_reference`).

### 1.1.5.2 Response-Format und Fehlerbehandlung

Die API liefert für alle Requests ein Datenpaket in folgenden JSON-Form zurück:

```
{
  "errors": [ ...errors ],
  "data": { ...data }
}
```

In `data` sind die eigentlichen Informationen in Antwort auf den jeweiligen Request enthalten. `errors` enthält bei Bearbeitung des Requests aufgetretene Fehler. Jedes Element in `errors` hat folgende Struktur:

```
{
  "message": String,
  "mapping": String | null,
  "code": String
}
```

Attribut	Typ	Inhalt
message	String	Die Nachrichteneigenschaft enthält eine für Menschen lesbare englische Beschreibung des Fehlers. Die Hauptzielgruppe ist der Programmierer, der mit der API arbeitet. Die Fehlerinformation sollte ihn auf dem Weg zur Behebung des Fehlers unterstützen.
mapping	String / null	Die Zuordnungseigenschaft enthält einen gültigen JavaScript-Ausdruck als Zeichenfolge, die die Eigenschaft des übermittelten oder bereitgestellten Objekts auswertet, bei dem der Fehler aufgetreten ist. Die Zuordnung kann null sein, wenn keine solche Verbindung hergestellt werden kann. Wenn ein Ausdruck wie <code>foo[1].bar</code> festgelegt ist, verweist <code>bar</code> auf die Eigenschaft im zweiten Element eines Arrays mit dem Namen <code>foo</code> , das sich im Stammobjekt befindet.
code	String	Der Code ist eine Zeichenfolge, die eine Fehlerklassenhierarchie darstellt, in der jede Fehlerklasse durch einen Punkt begrenzt ist. Dies ist hilfreich, wenn Sie mit wenigen Fehlerübersetzungen beginnen und zu einem späteren Zeitpunkt mit den Beschreibungen genauer werden möchten. Anstatt einen Code wie <code>LPLC.validation.non_negative_float</code> zu übersetzen, können Sie <code>LPLC.validation</code> einfach mit „Bitte überprüfen Sie Ihre Eingabe“ übersetzen und später eine Übersetzung für Float-Fehler hinzufügen. Falls eine API-Anfrage Fehlercodes zurückliefern kann, sind diese zu jeder Abfrage dokumentiert.

Fehlercodes werden für jede der folgenden Resource, siehe 1.2 REST-API Resources, für die Schreibbefehle (`POST` oder `PUT`) dokumentiert, die folgenden sind jedoch die häufigsten:

```
LPLC.validation
LPLC.validation.missing_input
LPLC.validation.readonly
LPLC.validation.non_negative_float
LPLC.validation.positive_integer
LPLC.validation.smaller_integer
LPLC.validation.single_character
LPLC.validation.string
LPLC.validation.boolean
LPLC.format.encoding.utf8
LPLC.format.malformed.json
```

### 1.1.5.3 Schaltausgänge, Trigger und Haltezeit

Jede Abtastperiode endet mit der Auswahl des am besten geeigneten Matchers ("Gruppe von Farben "). Dieser Matcher wird unter bestimmten Bedingungen auf die Schaltausgänge angewendet. Relevante Konfiguration für dieses Verhalten sind die folgenden Einstellungen:

- optional ausgelöste Aktualisierung der Schaltausgänge (siehe action-triggers AXXX)
- Matcher-Attribute *hold\_time* und *reset\_output\_after\_hold\_time\_expired*

Die folgenden Situationen und Aktionen werden in der folgenden Verhaltensspezifikation verwendet:

- *no active hold time*: Das Attribut *hold\_time* des zuletzt erkannten Matchers (Farbgruppe) war Null. Dies bedeutet, dass keine Haltezeit konfiguriert ist. Es liegt keine Haltezeit an, bis eine andere Farbgruppe mit einer Haltezeit ungleich Null erkannt wird.
- *hold time is expired*: Der zuletzt erkannte Matcher hatte eine Haltezeit größer als Null, aber diese Haltezeit ist seit dem Erkennen dieser Farbgruppe verstrichen. Somit war zuvor eine Haltezeit aktiv, die jedoch in der Zwischenzeit abgelaufen ist.
- *reset\_output\_after\_hold\_time\_expired* is on/off: Wenn ein Matcher erkannt wird (siehe unten), wird das Attribut *reset\_output\_after\_hold\_time\_expired* gespeichert, bis ein anderer Farbgruppe erkannt wird. Der Ein / Aus-Zustand bezieht sich auf diesen gespeicherten Wert.
- *unchanged detected matcher*: Der zuletzt erkannte Matcher und der aktuell erkannte Matcher sind identisch.
- *new detected matcher*: Der zuletzt erkannte Matcher und der aktuell erkannte Matcher sind nicht identisch.
- *action do nothing*: Die Schaltausgänge und die aktuell gespeicherten Haltezeiteinstellungen bleiben unverändert.
- *action apply new matcher*: Merken Sie sich die Matcher-Attribute *hold\_time* und *reset\_output\_after\_hold\_time\_expired* als aktuelle Haltezeiteinstellungen und stellen Sie die Schaltausgänge wie im *output\_pattern* des Matchers angegeben ein.
- *action apply 'no match'*: Merken Sie sich das Erkennungsprofilattribut *non\_matching\_hold\_time* als aktuelle Haltezeit. Merken Sie sich *false* für *reset\_output\_after\_hold\_time\_expired*. Stellen Sie die Schaltausgänge wie im Erkennungsprofilattribut *non\_matching\_output* angegeben ein.

Das Verhalten mit und ohne ausgelöste Aktualisierungen von Schaltausgängen unterscheidet sich erheblich. Daher werden beide Situationen im Folgenden separat angegeben.

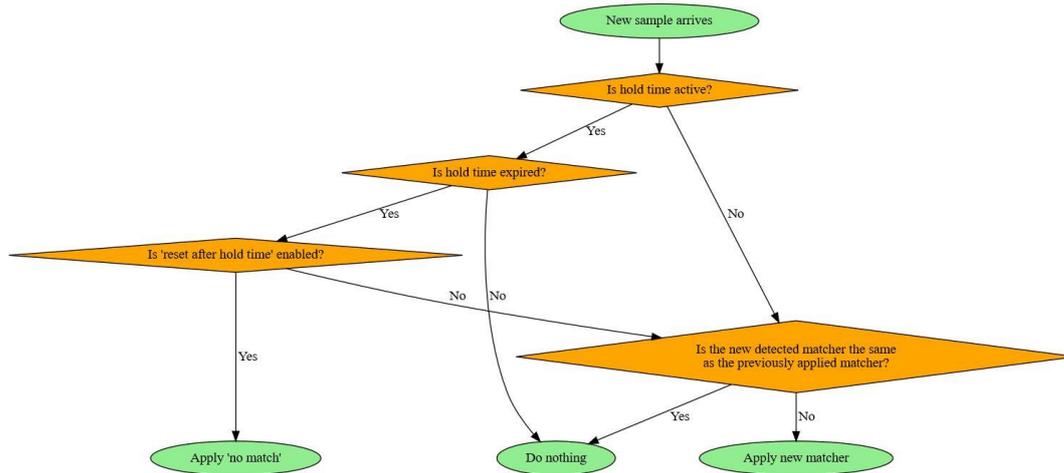
#### Deaktivierte getriggerte Aktualisierung der Schaltausgänge

Getriggerte Aktualisierungen sind deaktiviert, wenn keines der Auslöseereignisse (z. B. eine ansteigende Flanke des ersten Schalteingangs IN0) mit der Aktion *enable\_switching\_output* konfiguriert wurde (siehe [/api/actions](#)).

In der folgenden Tabelle ist das spezifische Verhalten basierend auf den derzeit aktiven Haltezeiteinstellungen und abhängig vom aktuell erkannten Matcher aufgeführt.

Derzeit aktive Haltezeiteinstellungen	Aktion für unveränderten erkannten Matcher	Aktion für neu erkannten Matcher
Keine aktive Haltezeit	Keine Veränderung des Ausgangsverhalten	Neue Farbgruppe ausgeben
Haltezeit ist abgelaufen; <i>reset_output_after_hold_time_expired</i> ist aus	Keine Veränderung des Ausgangsverhalten	Neue Farbgruppe ausgeben
Haltezeit ist abgelaufen; <i>reset_output_after_hold_time_expired</i> ist an	Ausgang für Farbe nicht erkannt wird ausgegeben	Ausgang für Farbe nicht erkannt wird ausgegeben
Haltezeit ist nicht abgelaufen	Keine Veränderung des Ausgangsverhalten	Keine Veränderung des Ausgangsverhalten

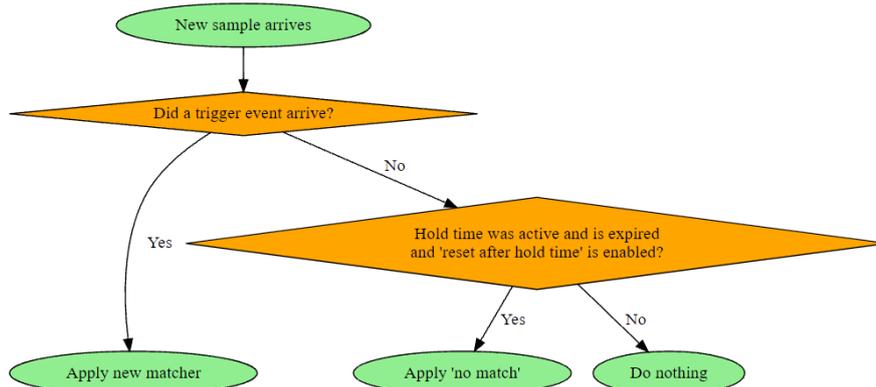
Das folgende Flussdiagramm zeigt die Entscheidungen und Aktionen beim Anwenden eines Matchers. Es ist eine andere Darstellung der obigen Tabelle.



### Aktiviere getriggerte Aktualisierung der Schaltausgänge

Getriggerte Aktualisierungen werden aktiviert, wenn eines der Trigger-Ereignisse (z.B. eine ansteigende Flanke des ersten Schalteingangs IN0) mit der Aktion *enable\_switching\_output* konfiguriert wird (siehe `/api/actions`).

Das folgende Flussdiagramm zeigt die Entscheidungen und Aktionen beim Erkennen eines Matchers.



#### 1.1.5.4 Websockets

Mit dem von der API bereitgestellten WebSocket können hochgradig interaktive Frontends, wie die Bedienungswebseiten für die Sensoren entwickelt werden, die funktionieren, ohne die Endpunkte der REST-ähnlichen „Resources“ abzufragen.

Neben dem stream von samples sendet der Controller auch Informationen über Ereignisse wie neue eingelernte Farben oder die Änderung der Sensorkonfiguration über den WebSocket.

Der WebSocket wurde als Ergänzung zu den „Resources“ und nicht als Ersatz entwickelt. Die primäre Schnittstelle zur Steuerung des Sensors sind und bleiben die unten aufgeführten „Resources“.

Anforderungen an einen der Endpunkte unter `/websocket` setzen das Sitzungszeitlimit nicht zurück.

### Überblick

Sie können auf den WebSocket über den Pfad `/websocket/notifications/websocket` zugreifen. Die auf diesem Kanal übertragenen Daten ähneln einem stream von Informationen. Um relevante Pakete leicht zu identifizieren, werden alle über den Socket übertragene Nutzdaten in eine JSON-Form eingekapselt, das die folgenden Eigenschaften enthält:

colorSENSOR CFO

- id

- Die eindeutige Kennung dieses bestimmten WebSocket-Pakets

- source

- Die Quelle, von der dieses Paket stammt, oder der Grund für seine Übertragung (z.B. `detection_profile.matcher`)

- timestamp

- Die Sensoren sind betriebsbereit, wenn das Paket über den WebSocket gesendet wird.

- payload

Objekt mit einem `event` Merkmal (mit Werten wie `changed` oder `created`), und einer optionalen `uuid` und `data` Attribut, das das beschriebene Objekt `event` von `source` darstellt.

Eines der Felder `added_items`, `changed_items` und `removed_items` kann vorhanden sein, wenn sich die `source` auf eine Sammlung bezieht. In diesem Fall ist der Wert dieses Felds eine Liste von Bezeichnern, die zu den betroffenen Elementen der Sammlung gehören. Auf diese Weise können Clients ihr Datenmodell synchronisieren, ohne nach jeder sammlungsbezogenen Benachrichtigung die vollständige Erfassung anzufordern. Wenn eine sammlungsbezogene Benachrichtigung keines der oben genannten Felder enthält, ist der Umfang der Änderungen nicht spezifisch und daher kann ein vollständiger Abruf der Sammlung erforderlich sein.

### Fallback

Um ältere Browser zu unterstützen ist der `/websocket/notifications` Endpunkt kompatibel mit der SockJS Client-Bibliothek, welche die Fallback-Methoden wie XHR-Streaming, JSONP, Long Polling und andere implementiert haben.

#### 1.1.5.5 Networking und Discovery

##### Network discovery

Der Controller meldet sich über die folgenden Protokolle im lokalen Netzwerk an:

- zeroconf / avahi broadcasts
- SSDP

Das SSDP-Protokoll ermöglicht die Erkennung der Sensoren über die Windows-Netzwerkumgebung. Das zeroconf-Protokoll ermöglicht die Erkennung unter Linux, MacOS und Mobilgeräten.

##### Automatisch zugewiesene link-local Adresse

Der Sensor ist über seine explizit konfigurierten IPv4- und IPv6-Adressen sowie über seine automatisch konfigurierte verbindungslokale Adresse erreichbar. Diese Adresse gehört zum Subnetz "fe80::/10", wobei der lokale Adressteil auf der MAC-Adresse des Sensors basiert (siehe "EUI-64"). Die verbindungslokale Adresse des Sensors kann in allen Netzwerken unabhängig von der Sensorkonfiguration verwendet werden. Somit bietet es unter allen Umständen eine stabile Adresse.

Link-local Adressen müssen im Allgemeinen mit der lokalen Netzwerkkennung versehen werden.

##### Beispiele

- Windows: `fe80::1234:56ff:fe78:90ab%0`
- Linux / MacOS / Android: `fe80::1234:56ff:fe78:90ab%eth0`

Die Netzwerkkennung (Suffix nach "%") in den obigen Beispielen muss an die lokale Konfiguration des an den Sensor angeschlossenen Geräts angepasst werden.

## 1.2 Resources

### 1.2.1 Controller Eigenschaften

Einige konstante Eigenschaften beschreiben den einzelnen Controller selbst und enthalten Informationen zu Modell und Hersteller.

#### 1.2.1.1 Controller Eigenschaften abrufen

**GET** / device

Die folgenden unveränderlichen Informationen werden vom Controller abgerufen.

Antwort

Code	Body	JSON-Objekt	
200	Properties (object)	DeviceInformation	
	<b>data</b> <u>DeviceInformation</u> , <b>required</b>	<b>id</b> DeviceSerialNumber, <b>required</b>	Serien Nummer
		<b>model_name</b> string, <b>required</b>	lesbarer Name des Controller Modells
		<b>model_key</b> string, <b>required</b>	eindeutige ID des Controller modells
		<b>variant</b> any of string or null, <b>required</b>	zeigt eine spezielle Option eines Controller Modells an
		<b>vendor_key</b> DeviceVendorKey, <b>required</b>	Eindeutiger Schlüssel, der den Hersteller identifiziert, der dieses Gerät vertreibt
		<b>vendor_name</b> DeviceVendorName, <b>required</b>	Name des Herstellers dieses Geräts
		<b>device_id</b> DeviceSerialNumber, optional, <b>Deprecated</b>	Veraltet: Verwenden Sie stattdessen "id" (wird in zukünftigen Versionen verschwinden)
		<b>model</b> string, optional, <b>Deprecated</b>	Veraltet: Verwenden Sie stattdessen "model_name" (wird in zukünftigen Versionen verschwinden)
		<b>vendor</b> DeviceVendorName, optional, <b>Deprecated</b>	Veraltet: Verwenden Sie stattdessen "vendor_name"
4XX	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> string, optional	maschinenlesbarer eindeutiger Fehlercode
		<b>mapping</b> string, optional	ein Verweis auf den Parameter, der den Fehler verursacht hat
		<b>message</b> string, optional	vom Menschen lesbare Fehlerbeschreibung

### Beispiele

```
{ "data":
  { "model_key": "me_cfo_200",
    "model": "CFO200",
    "vendor": "Micro-Epsilon Eltrotec GmbH",
    "device_id": "7455301813",
    "vendor_key": "eltrotec",
    "model_name": "CFO200",
    "vendor_name": "Micro-Epsilon Eltrotec GmbH",
    "id": "7455301813",
    "variant": "100"},
  "errors": [] }
```

## 1.2.2 Sensor

Mit diesen Befehlen fragen Sie alle Details der Controller Konfiguration und des Betriebs ab und können diese ändern.

### 1.2.2.1 Farbmesswerte auslesen

**GET** / sensor / samples

Gibt eine Liste von Messwerten aus der Farberkennung zurück.

Wenn keine zusätzlichen Abfrageparameter übergeben werden, enthält die Sammlung Beispiele aus der Vergangenheit. Sie können das Sample-Streaming mit dem `stream` Abfrageparameter aktivieren. In diesem Fall werden nur neue Messwerte zurückgegeben, sobald sie verfügbar sind.

Samples werden als Ringpuffer implementiert. Alte Proben werden aus der Sammlung entfernt, wenn neue Proben hinzugefügt werden.

#### Anfrage

Query Parameters	
<b>stream</b> Number, one of [0, 1], default: 0, optional	Steuert, ob der Stream-Modus aktiviert ist oder nicht.  Wenn das Streaming aktiviert ist, sendet nur der Controller ständig neu Aufgenommene Farbwerte an den Client. Die Anzahl der übertragenen Samples kann mit dem Abfrageparameter <code>stream_count</code> gesteuert werden.  Wenn das Streaming deaktiviert ist (dies ist die Standardeinstellung), werden die letzten Farbwerte inkl. Älterer Werte zurückgegeben.
<b>stream_count</b> Integer, default: 0, optional	Legt fest, wie viele Samples übertragen werden sollen, bevor die Verbindung beendet wird, wenn der Stream-Modus aktiviert wurde. Standardmäßig wird auf unbestimmte Zeit gestreamt.
<b>format</b> String, one of [json, csv], default: json, optional	Bestimmt das Ausgabeformat der Samples, wenn der Stream-Modus aktiviert wurde.  Wenn <code>csv</code> ausgewählt ist, sind die Spaltenüberschriften die erste übertragene Zeile. Die Header basieren auf der Standard-JSON-Darstellung und verwenden die Syntax, die auch von JavaScript verwendet wird. Bei den <code>representations.RGB[0]</code> würde sich der Wert für diesen Header auf das erste Element in der RGB-Darstellung beziehen.
<b>delimiter</b> String, default: ,, minimum length: 1, maximum length: 1, optional	Bestimmt das Trennzeichen zur Spaltenbegrenzung, wenn <code>csv</code> als Ausgabeformat ausgewählt wurde.  Wenn Sie ein Semikolon als Trennzeichen verwenden möchten, müssen Sie es zuerst per URL codieren ( <code>%3B</code> ). Andernfalls wird es als Trennzeichen für Abfrageparameter interpretiert.  Beachten Sie, dass Sie sich, obwohl Unicode-Zeichen von der API zugelassen werden, auf Ein-Byte-Zeichen beschränken sollten, da die meisten Tools keine Trennzeichen verwenden, die zwei oder mehr Bytes lang sind.

#### Antwort

Code	Body	JSON-Objekt	Inhalt
200	Properties (object)	Data	
	<b>data</b> object, required	<b>samples</b> Array of <a href="#">ColorDetectionResult</a> , required	ColorDetectionResult[]
		<b>uuid</b> <a href="#">UUID</a> (string), pattern: <code>^[a-f0-9-]+\$,</code> required, read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
		<b>timestamp</b> <a href="#">TimestampBackendUptime</a> (number), required	Der Zeitstempel (in Mikrosekunden angegeben) basiert auf der Betriebszeit des internen Backends des analogen Sensors. Unter bestimmten Bedingungen kann es auf Null zurückgesetzt werden.

Code	Body	JSON-Objekt		Inhalt		
			<b>corrected_color</b> <u>CorrectedColor</u> , required	Darstellung einer Farbe im Farbraum XYZ.		
				<b>values</b> Array of number, minimum items: 3, maximum items: 3, required	Ort in einem Farbraum	
			<b>transformed_color</b> <u>TransformedColor</u> , required	Eine Farbe, die durch eine Koordinate im Farbraum dargestellt wird. Die Array-Indizes der Eigenschaft values stimmen mit der Reihenfolge der Eigenschaft colorspace.axes des aktuell verwendeten Erkennungsprofils überein.		
				<b>values</b> Array of number, minimum items: 3, maximum items: 3, required	Ort in einem Farbraum	
			<b>representations</b> <u>ColorRepresentations</u> , required	Vorberechnete visuelle Darstellungen einer zum Rendern geeigneten Farbe		
				<b>RGB</b> Array of number, minimum items: 3, maximum items: 3, required	RGB-Farrray, das die Achsen r, g und b in dieser Reihenfolge darstellt. Werte sind Gleitkommazahlen zwischen 0 und 1.	
			<b>inputs</b> <u>InputsState</u> , required	Der Status aller Eingaben während eines bestimmten Zeitraums wird durch eine Liste möglicher Ereignisse in Kombination mit einem Booleschen Wert angegeben, der angibt, ob das angegebene Ereignis innerhalb des Zeitraums aufgetreten ist.		
				// boolean, required	Der boolesche Wert gibt an, ob das benannte Eingabeereignis in der letzten Periode aufgetreten ist.	
			<b>detection</b> <u>ColorMatchingResult</u> , required	Nach jeder Abtastperiode wird der abgerufene Farbwert mit den gespeicherten Farben (Detectables-Farbpositionen) verglichen. Farben (Detectables) werden ignoriert, wenn die Toleranzform der entsprechenden Farbgruppe (Matcher) die aktuelle Stichprobe nicht umfasst. Schließlich wird die geeignetste Farbe (Detectable) als Gewinner der Farbanpassungsoperation ausgewählt. Die entsprechende Farbgruppe (Matcher) bestimmt den Zustand des Sensors für die Dauer der nächsten Abtastperiode.		
				<b>matcher</b>	Veraltet: Verwenden Sie stattdessen	

Code	Body	JSON-Objekt		Inhalt		
				any of <u>UUID</u> (string) or null, optional, <b>Deprecated</b>	"selected_matcher_id"	
				<b>chosen_matcher_id</b> any of <u>UUID</u> (string) or null, <b>required</b>	eindeutige Kennung des ausgewählten Matchers	
				<b>distances</b> Array of any of number or null, <b>required</b>	Abstand zwischen der Farbposition der Probe und der nächstgelegenen Farbposition des ausgewählten Matchers entlang der drei Achsen des Farbraums. Das Array enthält drei Nullwerte, wenn für das aktuelle Farbmuster kein geeigneter Matcher gefunden wurde.	
				<b>output_pattern</b> <u>CurrentSwitchingOutputsState</u> , <b>required</b>	Derzeit aktiver Status der Schaltausgänge. Beachten Sie, dass dies von den angegebenen Ausgangszuständen des aktuell besten Matchers abweichen kann, da Einstellungen wie ausgelöste Eingabe oder Haltezeit den Aktualisierungsprozess für die Schaltausgänge beeinflussen.	
					<b>states</b> Array of any of boolean or null, <b>required</b>	Liste der True / False-Werte, die den aktuellen Status der Schaltausgänge beschreiben
			<b>signal_level</b> number, <b>required</b>	Der Signalpegel zeigt die Verwendung des internen ADC-Abtastbereichs an		
	<b>errors</b> Array of Error, <b>required</b>	Error[]				
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode			
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte			
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung			

### 1.2.2.2 Letzte Farbmesswerte auslesen

**GET** / sensor / samples / current

Gibt die letzten Messwerte aus.

Seien Sie sich im Klaren, dass möglicherweise die gleichen Messwerte für nachfolgende Anfragen ausgegeben werden wenn zwischenzeitlich keine neuen Messwerte eingegangen sind. Solang der Sensor auto-gain durchführt oder übersättigt ist, wird er den letzten gültigen Messwert ausgegeben der verarbeitet wurde.

Das Ergebnis ist leer (`null`), während der Sensor eine Änderungsanforderung der Konfiguration verarbeitet.

Antwort

Code	Body	JSON-Objekt	Inhalt
200	Properties (object)		
	<b>data</b> Any of <a href="#">ColorDetectionResult</a> or <code>null</code> , <b>required</b>	ColorDetection-Result	
		<b>uuid</b> <a href="#">UUID</a> (string), pattern: <code>^[a-f0-9-]+\$, <b>required</b>, read-only</code>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
		<b>timestamp</b> <a href="#">TimestampBackendUptime</a> (number), <b>required</b>	Der Zeitstempel (in Mikrosekunden angegeben) basiert auf der Betriebszeit des internen Backends des analogen Sensors. Unter bestimmten Bedingungen kann es auf Null zurückgesetzt werden
		<b>corrected_color</b> <a href="#">CorrectedColor</a> , <b>required</b>	Darstellung einer Farbe im Farbraum XYZ
			<b>values</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>
			Ort in einem Farbraum
		<b>transformed_color</b> <a href="#">TransformedColor</a> , <b>required</b>	Eine Farbe, die durch eine Koordinate im Farbraum dargestellt wird. Die Array-Indizes der Eigenschaft values stimmen mit der Reihenfolge der Eigenschaft colorspace.axes des aktuell verwendeten Erkennungsprofils überein.
			<b>values</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>
			Ort in einem Farbraum
		<b>representations</b> <a href="#">ColorRepresentations</a> , <b>required</b>	Vorberechnete visuelle Darstellungen einer zum Rendern geeigneten Farbe
			<b>RGB</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>
			RGB-Farrray, das die Achsen r, g und b in dieser Reihenfolge darstellt. Werte sind Gleitkommazahlen zwischen 0 und 1.
		<b>inputs</b> <a href="#">InputsState</a> , <b>required</b>	Der Status aller Eingaben während eines bestimmten Zeitraums wird durch eine Liste möglicher Ereignisse in Kombination mit einem Booleschen Wert angegeben, der angibt, ob das angegebene Ereignis innerhalb des Zeitraums aufgetreten ist.

Code	Body	JSON-Objekt		Inhalt	
			// boolean, <b>required</b>	Der boolesche Wert gibt an, ob das benannte Eingabeereignis in der letzten Periode aufgetreten ist.	
		<b>detection</b> <a href="#">ColorMatchingResult</a> , <b>required</b>	Nach jeder Abtastperiode wird der abgerufene Farbwert mit den gespeicherten Farben (DetectablesFarbpositionen) verglichen. Farben (Detectables) werden ignoriert, wenn die Toleranzform der entsprechenden Farbgruppe (Matcher) die aktuelle Stichprobe nicht umfasst. Schließlich wird die geeignetste Farbe (Detectable) als Gewinner der Farbanpassungsoperation ausgewählt. Die entsprechende Farbgruppe (Matcher) bestimmt den Zustand des Sensors für die Dauer der nächsten Abtastperiode.		
			<b>matcher</b> any of <a href="#">UUID</a> (string) or null, optional, <b>Deprecated</b>	Veraltet: Verwenden Sie stattdessen "selected_matcher_id"	
			<b>chosen_matcher_id</b> any of <a href="#">UUID</a> (string) or null, <b>required</b>	eindeutige Kennung des ausgewählten Matchers	
			<b>distances</b> Array of any of number or null, <b>required</b>	Abstand zwischen der Farbposition der Probe und der nächstgelegenen Farbposition des ausgewählten Matchers entlang der drei Achsen des Farbraums. Das Array enthält drei Nullwerte, wenn für das aktuelle Farbmuster kein geeigneter Matcher gefunden wurde.	
			<b>output_pattern</b> <a href="#">CurrentSwitchingOutputsState</a> , <b>required</b>	Derzeit aktiver Status der Schaltausgänge. Beachten Sie, dass dies von den angegebenen Ausgangszuständen des aktuell besten Matchers abweichen kann, da Einstellungen wie ausgelöste Eingabe oder Haltezeit den Aktualisierungsprozess für die Schaltausgänge beeinflussen.	
				<b>states</b> Array of any of boolean or null, <b>required</b>	Liste der True / False-Werte, die den aktuellen Status der Schaltausgänge beschreiben
		<b>signal_level</b> number, <b>required</b>	Der Signalpegel zeigt die Verwendung des internen ADC-Abtastbereichs an		
	<b>errors</b> Array of Error, <b>required</b>	Error[]			
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode		
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte		
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung		

### 1.2.2.3 Alle Farbgruppen (Matchers) auslesen

**GET** / sensor / matchers

Farbgruppen (Matchers) beschreiben ein Farberkennungsergebnis. Eine Farbgruppe beinhaltet Details bezüglich der Eigenschaften der Schaltausgänge und eine Liste von Farben (Detectables).

Die einfachste Konfiguration eines Farbsensors könnte nur eine einzige Farbgruppe beinhalten, welche alle positiven Erkennungsergebnisse beinhaltet. Alle Messwerte welche nicht zu dieser Farbgruppe passen, würden ein Problem des beobachteten realen Prozesses angeben.

Eine erweiterende Verwendung von Matchern könnte zusätzlich eine Farbgruppe (Matcher) für die verschiedenen akzeptablen Hintergrundfarben zwischen realen Zielobjekten (z.B. die Farbe des Förderbandes) beinhalten. Somit geben die Schaltausgänge des Sensors an, ob ein positives, ein neutrales oder ein negatives Schaltsignal erkannt wurde.

Selbstverständlich kann jede Farbgruppe (Matcher) auch einfach nur eine Farbposition beinhalten, um eine detaillierte Erkennung des Zielobjekts zu begünstigen.

Das `tolerance` Feld einer Farbgruppe (Matcher), beschreibt die Form und die Dimension des Teils des Farbraumes, welcher von der Farbgruppe (Matcher) abgedeckt wird. Nur Farbpositionen innerhalb dieses Raumes können zugeordnet werden. Eine Toleranz wird bestimmt durch eine `shape` und ein Lexikon aus `limits`. Beide Attribute müssen bestimmt werden. Ein leeres Lexikon aus `limits` wird als Standardrahmen für die angefragte Form interpretiert.

#### Anfrage

Query Parameters	
<b>profile_id</b> String, optional	Filter ColorMatchers nach der gegebenen <i>Detection Profile ID</i> . Nur ColorMatchers, welche Teil des gegebenen Erkennungsprofils (Detection Profile) sind werden erkannt.

#### Antwort

Code	Body	JSON-Objekt	Inhalt
200	Properties (object)		
	<b>data</b> Object, required	data	
		<b>matchers</b> Array of <a href="#">ColorMatcher</a> , required	ColorMatcher[]
		<b>uuid</b> <a href="#">UUID</a> (string), pattern: <code>^[a-f0-9-]+\$,</code> required, read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
		<b>alias</b> <a href="#">Alias</a> (integer), required, read-only	Ein numerischer Wert, welcher benutzt werden kann, um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.
		<b>name</b> String, required	Visuell lesbarer Name der Farbgruppe (matcher)
		<b>tolerance</b> Any of <a href="#">InfiniteColorTolerance</a> , <a href="#">SphereColorTolerance</a> , <a href="#">CylinderColorTolerance</a> or <a href="#">BoxColorTolerance</a> , required	Spezifikation einer geometrischen Form und ihrer Dimension in dem aktuellen Farbraum.

Code	Body	JSON-Objekt	Inhalt	
			InfiniteColorTolerance	
			<b>limits</b> Object, <b>required</b>	limits
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via /api/sensor/capabilities.
			SphereColorTolerance	
			<b>limits</b> Object, <b>required</b>	limits <b>radius</b> Number, <b>required</b>
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via /api/sensor/capabilities.
			CylinderColorTolerance	
			<b>limits</b> Object, <b>required</b>	limits <b>radius</b> Number, <b>required</b> <b>half_height</b> Number, <b>required</b>
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via /api/sensor/capabilities.
			BoxColorTolerance	
			<b>limits</b> Object, <b>required</b>	limits <b>half_edges</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via /api/sensor/capabilities.
		<b>output_pattern</b> WantedSwitchingOutputsState, <b>required</b>	Die Kombination von tristate Werten beschreibt einen logischen Zustand der Schaltausgänge des Sensors. Die Zustände true oder false veranlassen den Ausgang anzusteigen oder zu sinken. Der Zustand null lässt den vorherigen Zustand der Ausgabe unverändert.	
			<b>uuid</b> UUID (string), pattern: ^[a-f0-9-]+\$, <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
			<b>states</b> Array of any of boolean or null, <b>required</b>	Liste von True/False/Null Werten beschreibt den gesuchten Zustand der Schaltausgänge.
		<b>hold_time</b> HoldTime (number), maximum: 3153600000, <b>required</b>	Minimum Laufzeit (in Sekunden) der Ausgabekonfiguration einer Farbgruppe, bei Anwendung nach der Erkennung.	
		<b>reset_output_after_hold_time_expired</b> Boolean, default: false, <b>required</b>	Steuert ob die Ausgabe zurückgesetzt werden soll, nachdem die Haltezeit überschritten wurde. Dies ist hilfreich, wenn Sie nur durch auslösende Eingaben abfragen und wünschen im Nachhinein die Ausgaben zurückzusetzen.	

Code	Body	JSON-Objekt		Inhalt
			<b>signal_color</b> Any of string or null, <b>required</b>	Ein benutzerdefinierter Farbname. Wie und welche Farbe angezeigt wird, wird durch den Kunden bestimmt.
	<b>errors</b> Array of Error, <b>required</b>	Error []		
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode	
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte	
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung	

### 1.2.2.4 Erstelle eine neue Farbgruppe (Matcher)

**POST** / sensor / matchers

Speichert eine neue Farbgruppe (Matcher) auf dem Sensor. Um darauf Farben hinzuzufügen, benutzen Sie den Endpunkt `/api/sensor/detectables`.

#### Anfrage

Body	JSON-Objekt	Inhalt	
<b>Properties</b> (ColorMatcher)			
	<b>uuid</b> UUID (string), pattern: ^ [a-f0-9-]+\$, <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
	<b>alias</b> Alias (integer), <b>required</b> , read-only	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.	
	<b>name</b> String, <b>required</b>	Visuell lesbarer Name der Farbgruppe (matcher)	
	<b>tolerance</b> Any of InfiniteColorTolerance, SphereColorTolerance, CylinderColorTolerance or BoxColorTolerance, <b>required</b>	Spezifikation einer geometrischen Form und ihrer Dimension in dem aktuellen Farbraum.	
		InfiniteColorTolerance	
		<b>limits</b> Object, <b>required</b>	Limits
		<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
		SphereColorTolerance	
		<b>limits</b> Object, <b>required</b>	Limits <b>radius</b> Numer, <b>required</b>
		<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unter-

Body	JSON-Objekt	Inhalt	
			stützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
		CylinderColorTolerance	
		<b>limits</b> Object, <b>required</b>	Limits <b>radius</b> Number, <b>required</b> <b>half_height</b> Number, <b>required</b>
		<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
		BoxColorTolerance	
		<b>limits</b> Object, <b>required</b>	Limits <b>half_edges</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>
		<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
	<b>output_pattern</b> WantedSwitchingOutputsState, <b>required</b>	Die Kombination von tristate Werten beschreibt einen logischen Zustand der Schaltausgänge des Sensors. Die Zustände <code>true</code> oder <code>false</code> veranlassen den Ausgang anzusteigen oder zu sinken. Der Zustand <code>null</code> lässt den vorherigen Zustand der Ausgabe unverändert.	
		<b>uuid</b> UUID (string), pattern: <code>^[a-f0-9-]+\$, <b>required</b>, read-only</code>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
		<b>states</b> Array of any of boolean or null, <b>required</b>	Liste von True/False/Null Werten beschreibt den gesuchten Zustand der Schaltausgänge.
	<b>hold_time</b> HoldTime (number), maximum: 3153600000, <b>required</b>	Minimum Laufzeit (in Sekunden) der Ausgabekonfiguration einer Farbgruppe, bei Anwendung nach der Erkennung.	
	<b>reset_output_after_hold_time_expired</b> Boolean, default: false, <b>required</b>	Steuert ob die Ausgabe zurückgesetzt werden soll, nachdem die Haltezeit überschritten wurde. Dies ist hilfreich, wenn Sie nur durch auslösende Eingaben abfragen und wünschen im Nachhinein die Ausgaben zurückzusetzen.	
	<b>signal_color</b> Any of string or null, <b>required</b>	Ein benutzerdefinierter Farbname. Wie und welche Farbe angezeigt wird, wird durch den Kunden bestimmt.	

## Beispiele

```
{
  "uuid": "9ffaa31f-8011-44f5-bb2a-f91e4be50764",
  "alias": 6,
  "name": "clean bottle cap",
  "tolerance": {
    "limits": {
      "radius": 2,
      "half_height": 4
    },
    "shape": "cylinder"
  }
}
```

```

},
"output_pattern": {
  "uuid": "1adc74e2-96ac-4761-b9e6-2d93e02d9244",
  "states": [
    true,
    false,
    false
  ]
},
"hold_time": 0,
"reset_output_after_hold_time_expired": false,
"signal_color": null
}

```

Antwort

Code	Body	JSON-Objekt		Inhalt	
200 400	Properties (object)				
	<b>data</b> <u>ColorMatcher</u> , <b>required</b>	Eine Farbgruppe vertritt ein ausgezeichnetes Erkennungsergebnis und das gewollte Verhalten des Sensors, wann immer er angetroffen wird.			
			ColorMatcher[]		
			<b>uuid</b> <u>UUID</u> (string), pattern: <code>^[a-f0-9-]+\$, <b>required</b>, read-only</code>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
			<b>alias</b> <u>Alias</u> (integer), <b>required</b> , read-only	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.	
			<b>name</b> String, <b>required</b>	Visuell lesbarer Name der Farbgruppe (matcher)	
			<b>tolerance</b> Any of InfiniteColorTolerance, SphereColorTolerance, CylinderColorTolerance or BoxColorTolerance, <b>required</b>	Spezifikation einer geometrischen Form und ihrer Dimension in dem aktuellen Farbraum.	
				InfiniteColorTolerance	
			<b>limits</b> Object, <b>required</b>		limits
			<b>shape</b> <u>ToleranceShapeName</u> (string), <b>required</b>		Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
				SphereColorTolerance	
			<b>limits</b>		limits

Code	Body	JSON-Objekt	Inhalt	
			Object, <b>required</b>	<b>radius</b> Numer, <b>required</b>
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
			CylinderColorTolerance	
			<b>limits</b> Object, <b>required</b>	<b>limits</b> <b>radius</b> Number, <b>required</b> <b>half_height</b> Number, <b>required</b>
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
			BoxColorTolerance	
			<b>limits</b> Object, <b>required</b>	<b>limits</b> <b>half_edges</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
		<b>output_pattern</b> WantedSwitchingOutputsState, <b>required</b>	Die Kombination von tristate Werten beschreibt einen logischen Zustand der Schaltausgänge des Sensors. Die Zustände <code>true</code> oder <code>false</code> veranlassen den Ausgang anzusteigen oder zu sinken. Der Zustand <code>null</code> lässt den vorherigen Zustand der Ausgabe unverändert.	
			<b>uuid</b> UUID (string), pattern: <code>^[a-f0-9-]+\$, <b>required</b>, read-only</code>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
			<b>states</b> Array of any of boolean or null, <b>required</b>	Liste von True/False/Null Werten beschreibt den gesuchten Zustand der Schaltausgänge.
		<b>hold_time</b> HoldTime (number), maximum: 3153600000, <b>required</b>	Minimum Laufzeit (in Sekunden) der Ausgabekonfiguration einer Farbgruppe, bei Anwendung nach der Erkennung.	
		<b>reset_output_after_hold_time_expired</b> Boolean, default: false, <b>required</b>	Steuert ob die Ausgabe zurückgesetzt werden soll, nachdem die Haltezeit überschritten wurde. Dies ist hilfreich, wenn Sie nur durch auslösende Eingaben abfragen und wünschen im Nachhinein die Ausgaben zurückzusetzen.	
		<b>signal_color</b> Any of string or null, <b>required</b>	Ein benutzerdefinierter Farbname. Wie und welche Farbe angezeigt wird, wird durch den Kunden bestimmt.	
	<b>errors</b>	Error []		

Code	Body	JSON-Objekt		Inhalt	
	Array of Error, required				
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode		
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte		
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung		
<b>Könnte folgende Fehlercodes ausgeben</b>					
LPLC.validation.collection_size_exceeded					

### 1.2.2.5 Entferne mehrere oder alle Farbgruppen (Matchers)

**DELETE** / sensor / matchers

Entferne eine Auswahl an Farbgruppen (Matchers), entweder basierend auf einem gegebenem Filterkriterium (falls unterstützt für diese Auswahl) oder entferne alle Farbgruppen (Matchers) von der Sammlung.

Alle Lösch-Anfragen resultieren in einer leeren Erfolgsrückmeldung (204). Dies ist sogar gültig für eine ungefilterte **DELETE**-Anfrage entgegen einer leeren Sammlung oder für eine gefilterte **DELETE**-Anfrage entgegen einer Sammlung ohne Farbgruppen (Matchers) die dem Filter entsprechen.

Antwort

Code	
204	Die leere Antwort weist auf Erfolg hin.

### 1.2.2.6 Lese Farbgruppendetails (Matcher details) aus

**GET** / sensor / matchers / {itemId}

Gibt die Konfiguration einer Farbgruppe aus.

Anfrage

Path Variables
<b>itemId</b> String, required

Antwort

Code	Body	JSON-Objekt		Inhalt	
200 400	Properties (object)				
	<b>data</b> <u>ColorMatcher</u> , required	Eine Farbgruppe vertritt ein ausgezeichnetes Erkennungsergebnis und das gewollte Verhalten des Sensors, wann immer er angetroffen wird.			

Code	Body	JSON-Objekt	Inhalt	
			ColorMatcher[]	
			<b>uuid</b> UUID (string), pattern: <code>^[a-f0-9-]+\$, required, read-only</code>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
			<b>alias</b> Alias (integer), <b>required, read-only</b>	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.
			<b>name</b> String, <b>required</b>	Visuell lesbarer Name der Farbgruppe (matcher)
			<b>tolerance</b> Any of InfiniteColorTolerance, SphereColorTolerance, CylinderColorTolerance or BoxColorTolerance, <b>required</b>	Spezifikation einer geometrischen Form und ihrer Dimension in dem aktuellen Farbraum.
				InfiniteColorTolerance
			<b>limits</b> Object, <b>required</b>	limits
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
				SphereColorTolerance
			<b>limits</b> Object, <b>required</b>	limits <b>radius</b> Numer, <b>required</b>
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
				CylinderColorTolerance
			<b>limits</b> Object, <b>required</b>	limits <b>radius</b> Number, <b>required</b> <b>half_height</b> Number, <b>required</b>
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
				BoxColorTolerance
			<b>limits</b> Object, <b>required</b>	limits <b>half_edges</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unter-

Code	Body	JSON-Objekt		Inhalt	
					stützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
			<b>output_pattern</b> WantedSwitchingOutputsState, <b>required</b>	Die Kombination von tristate Werten beschreibt einen logischen Zustand der Schaltausgänge des Sensors. Die Zustände <code>true</code> oder <code>false</code> veranlassen den Ausgang anzusteigen oder zu sinken. Der Zustand <code>null</code> lässt den vorherigen Zustand der Ausgabe unverändert.	
				<b>uuid</b> UUID (string), pattern: <code>^[a-f0-9-]+\$, <b>required</b>, read-only</code>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
				<b>states</b> Array of any of boolean or null, <b>required</b>	Liste von True/False/Null Werten beschreibt den gesuchten Zustand der Schaltausgänge.
			<b>hold_time</b> HoldTime (number), maximum: 3153600000, <b>required</b>	Minimum Laufzeit (in Sekunden) der Ausgabekonfiguration einer Farbgruppe, bei Anwendung nach der Erkennung.	
			<b>reset_output_after_hold_time_expired</b> Boolean, default: false, <b>required</b>	Steuert ob die Ausgabe zurückgesetzt werden soll, nachdem die Haltezeit überschritten wurde. Dies ist hilfreich, wenn Sie nur durch auslösende Eingaben abfragen und wünschen im Nachhinein die Ausgaben zurückzusetzen.	
			<b>signal_color</b> Any of string or null, <b>required</b>	Ein benutzerdefinierter Farbname. Wie und welche Farbe angezeigt wird, wird durch den Kunden bestimmt.	
	<b>errors</b> Array of Error, <b>required</b>	Error []			
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode		
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte		
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung		
<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item					

### 1.2.2.7 Löschen einer Farbgruppe (Matcher)

**DELETE** / sensor / matchers / {itemId}

Löscht die Farbgruppe und alle dazugehörigen Farben (detectables).

Anfrage

<b>Path variables</b>
<b>itemId</b> String, <b>required</b>

Antwort

<b>Code</b>	
204	Die leere Antwort weist auf Erfolg hin
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item

1.2.2.8 Aktualisiere die Farbgruppenkonfiguration (Matcher Configuration)

**PUT** / sensor / matchers / {itemId}

Aktualisiere die Farbgruppenkonfiguration (Matcher Konfiguration).

Anfrage

<b>Path Variables</b>
<b>itemId</b> String, required

Body	JSON-Objekt	Inhalt
<b>Properties</b> (Color Matcher)		
	<b>uuid</b> UUID (string), pattern: ^ [a-f0-9-]+\$, <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
	<b>alias</b> Alias (integer), <b>required</b> , read-only	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.
	<b>name</b> String, <b>required</b>	Visuell lesbarer Name der Farbgruppe (matcher)
	<b>tolerance</b> Any of InfiniteColorTolerance, SphereColorTolerance, CylinderColorTolerance or BoxColorTolerance, <b>required</b>	Spezifikation einer geometrischen Form und ihrer Dimension in dem aktuellen Farbraum.
		InfiniteColorTolerance
	<b>limits</b> Object, <b>required</b>	limits
	<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via /api/sensor/capabilities .
		SphereColorTolerance
	<b>limits</b> Object, <b>required</b>	limits <b>radius</b> Numer, <b>required</b>
	<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via /api/sensor/capabilities .
		CylinderColorTolerance
	<b>limits</b> Object, <b>required</b>	limits <b>radius</b> Number, <b>required</b> <b>half_height</b> Number, <b>required</b>

Body	JSON-Objekt	Inhalt	
		<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
		BoxColorTolerance	
		<b>limits</b> Object, <b>required</b>	<b>limits</b> <b>half_edges</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>
		<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
	<b>output_pattern</b> WantedSwitchingOutputsState, <b>required</b>	Die Kombination von tristate Werten beschreibt einen logischen Zustand der Schaltausgänge des Sensors. Die Zustände <code>true</code> oder <code>false</code> veranlassen den Ausgang anzusteigen oder zu sinken. Der Zustand <code>null</code> lässt den vorherigen Zustand der Ausgabe unverändert.	
		<b>uuid</b> <u>UUID</u> (string), pattern: <code>^[a-f0-9-]+\$,</code> <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
		<b>states</b> Array of any of boolean or null, <b>required</b>	Liste von True/False/Null Werten beschreibt den gesuchten Zustand der Schaltausgänge.
	<b>hold_time</b> HoldTime (number), maximum: 3153600000, <b>required</b>	Minimum Laufzeit (in Sekunden) der Ausgabekonfiguration einer Farbgruppe, bei Anwendung nach der Erkennung.	
	<b>reset_output_after_hold_time_expired</b> Boolean, default: <code>false</code> , <b>required</b>	Steuert ob die Ausgabe zurückgesetzt werden soll, nachdem die Haltezeit überschritten wurde. Dies ist hilfreich, wenn Sie nur durch auslösende Eingaben abfragen und wünschen im Nachhinein die Ausgaben zurückzusetzen.	
	<b>signal_color</b> Any of string or null, <b>required</b>	Ein benutzerdefinierter Farbname. Wie und welche Farbe angezeigt wird, wird durch den Kunden bestimmt.	

## Beispiele

```
{
  "uuid": "9ffaa31f-8011-44f5-bb2a-f91e4be50764",
  "alias": 6,
  "name": "clean bottle cap",
  "tolerance": {
    "limits": {
      "radius": 2,
      "half_height": 4
    },
    "shape": "cylinder"
  },
  "output_pattern": {
    "uuid": "1adc74e2-96ac-4761-b9e6-2d93e02d9244",
    "states": [
      true,
      false,
      false
    ]
  },
  "hold_time": 0,
  "reset_output_after_hold_time_expired": false,
  "signal_color": null
}
```

}

Antwort

Code	Body	JSON-Objekt		Inhalt		
200 400 404	Properties (object)					
	<b>data</b> <u>ColorMatcher</u> , <b>required</b>	Eine Farbgruppe vertritt ein ausgezeichnetes Erkennungsergebnis und das gewollte Verhalten des Sensors, wann immer er ange- troffen wird.				
			ColorMatcher			
			<b>uuid</b> <u>UUID</u> (string), pattern: <code>^[a-f0-9-]+\$, <b>required</b>, read-only</code>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8		
			<b>alias</b> <u>Alias</u> (integer), <b>required</b> , read-only	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.		
			<b>name</b> <u>String</u> , <b>required</b>	Visuell lesbarer Name der Farbgruppe (matcher)		
			<b>tolerance</b> Any of <u>InfiniteColorTolerance</u> , <u>SphereColorTolerance</u> , <u>CylinderColorTolerance</u> or <u>BoxColorTolerance</u> , <b>required</b>	Spezifikation einer geometrischen Form und ihrer Dimension in dem aktuellen Farbraum.		
				InfiniteColorTolerance		
				<b>limits</b> Object, <b>required</b>	limits	
				<b>shape</b> <u>ToleranceShapeName</u> (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .	
				SphereColorTolerance		
				<b>limits</b> Object, <b>required</b>	limits <b>radius</b> Numer, <b>required</b>	
				<b>shape</b> <u>ToleranceShapeName</u> (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranz-	

Code	Body	JSON-Objekt	Inhalt		
					form kann abgerufen werden via /api/sensor/capabilities .
				CylinderColorTolerance	
			<b>limits</b> Object, <b>required</b>	<b>limits</b> <b>radius</b> Number, <b>required</b> <b>half_height</b> Number, <b>required</b>	
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via /api/sensor/capabilities .	
				BoxColorTolerance	
			<b>limits</b> Object, <b>required</b>	<b>limits</b> <b>half_edges</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via /api/sensor/capabilities .	
		<b>output_pattern</b> WantedSwitchingOutputsState, <b>required</b>	Die Kombination von tristate Werten beschreibt einen logischen Zustand der Schaltausgänge des Sensors. Die Zustände <code>true</code> oder <code>false</code> veranlassen den Ausgang anzusteigen oder zu sinken. Der Zustand <code>null</code> lässt den vorherigen Zustand der Ausgabe unverändert.		
			<b>uuid</b> UUID (string), pattern: ^ [a-f0-9-]+\$, <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
			<b>states</b> Array of any of boolean or null, <b>required</b>	Liste von True/False/Null Werten beschreibt den gesuchten Zustand der Schaltausgänge.	
		<b>hold_time</b> HoldTime (number), maximum: 3153600000, <b>required</b>	Minimum Laufzeit (in Sekunden) der Ausgabekonfiguration einer Farbgruppe, bei Anwendung nach der Erkennung.		
		<b>reset_output_after_hold_time_expired</b> Boolean, default: false, <b>required</b>	Steuert ob die Ausgabe zurückgesetzt werden soll, nachdem die Haltezeit überschritten wurde. Dies		

Code	Body	JSON-Objekt		Inhalt		
				ist hilfreich, wenn Sie nur durch auslösende Eingaben abfragen und wünschen im Nachhinein die Ausgaben zurückzusetzen.		
			<b>signal_color</b> Any of string or null, <b>required</b>	Ein benutzerdefinierter Farbname. Wie und welche Farbe angezeigt wird, wird durch den Kunden bestimmt.		
	<b>errors</b> Array of Error, <b>required</b>	Error []				
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode			
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte			
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung			
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item					

### 1.2.2.9 Farben auslesen (Color Detectables)

**GET** / sensor / detectables

Die Farben (Detectables) beschreiben Positionen in dem aktuell ausgewählten Farbraum. Jede Farbe (Detectable) ist Teil einer Farbgruppe (Matcher). Jede Farbgruppe (Matcher) kann null oder mehr Farben (Detectables) enthalten.

Farben (Detectables) werden benutzt um die geeignetste Farbgruppe (Matcher) für eine Farbstichprobe zu bestimmen. Diese engste Übereinstimmung bestimmt das Ergebnis des Stichprobenzeitraums und somit das Verhalten des Sensors während des nächsten Stichprobenzeitraums.

Anfrage

Query Parameters	
<b>matcher_id</b> UUID (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	Filter Farben (detectables) nach der gegebenen Farbgruppen ID (Matcher ID). Nur Farben (detectables), welche Teil der gegebenen Farbgruppe (Matcher) sind werden ausgegeben.
<b>profile_id</b> String, optional	Filter Farben (ColorDetectables) nach der gegebenen Detection Profile ID. Nur Farben (ColorDetectables), welche Teil des gegebenen Erkennungsprofils (Detection Profile) sind werden ausgegeben.

Antwort

Code	Body	JSON-Objekt		Inhalt		
200	Properties (object)					
	<b>data</b> Object, <b>required</b>	<b>detectables</b> Array of Color-Detectable, <b>required</b>				
			ColorDetectable[]			
			<b>uuid</b> UUID (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8		

Code	Body	JSON-Objekt		Inhalt		
			<b>alias</b> <i>Alias</i> (integer), <b>required</b> , read-only	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.		
			<b>matcher_id</b> <i>UUID</i> (string), pattern: <code>^[a-f0-9-]+\$, <b>required</b>, read-only</code>	Referenz zu der Farbgruppe (Matcher), welche diese Farbe (Detectable) enthält.		
			<b>color</b> <i>TransformedColor</i> , <b>required</b>	Eine Farbe, welche durch Koordinaten im Farbraum angegeben wird. Die Array-Anzeichen des Besitzes des values entsprechen der Reihenfolge des Besitzes des color-space.axes des aktuellen Erkennungsprofils.		
				color		
				<b>values</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	Ort in einem Farbraum	
			<b>representations</b> <i>ColorRepresentations</i> , optional, read-only	Vorberechnete visuelle Darstellung einer geeigneten Farbe zur Wiedergabe		
				representations		
				<b>RGB</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	RGB-Farrray, das die Achsen r, g und b in dieser Reihenfolge darstellt. Werte sind Gleitkommazahlen zwischen 0 und 1.	
	<b>errors</b> Array of Error, <b>required</b>	Error[]				
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode			
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte			
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung			

### 1.2.2.10 Entferne mehrere oder alle Farben (ColorDetectables)

**DELETE** / sensor / detectables

Entferne eine Auswahl an Farben (ColorDetectables), entweder basierend auf einem gegebenem Filterkriterium (falls unterstützt für diese Auswahl) oder entferne alle Farben (ColorDetectables) von der Sammlung.

Alle Lösch-Anfragen resultieren in einer leeren Erfolgsrückmeldung (204). Dies ist sogar gültig für eine ungefilterte **DELETE**-Anfrage entgegen einer leeren Sammlung oder für eine gefilterte **DELETE**-Anfrage entgegen einer Sammlung ohne Farben (ColorDetectables) die dem Filter entsprechen.

Anfrage

Query Parameters	
<b>matcher_id</b> UUID (string), pattern: <code>^[a-f0-9-]+\$, required, read-only</code>	Entferne nur Farben (Detectables) mit der gegebenen Matcher ID.
<b>profile_id</b> String, optional	Filter Farben (ColorDetectables) nach der gegebenen Detection Profile ID. Nur Farben (ColorDetectables), welche Teil des gegebenen Erkennungsprofils (Detection Profile) sind werden ausgegeben.

Antwort

<b>Code</b>	
204	Die leere Antwort weist auf Erfolg hin.

1.2.2.11 Erstelle Farben (ColorDetectables)

**POST** / sensor / detectables

Erstelle eine neue Farbe (ColorDetectables).

Alle unterstützten Datenattribute in dem Body der Anfrage sind optional.

Anfrage

Body	JSON-Objekt		
<b>Properties</b> (ColorDetectable)	<b>uuid</b> UUID (string), pattern: <code>^[a-f0-9-]+\$, required, read-only</code>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
	<b>alias</b> Alias (integer), <b>required, read-only</b>	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.	
	<b>matcher_id</b> UUID (string), pattern: <code>^[a-f0-9-]+\$, required, read-only</code>	Referenz zu der Farbgruppe (Matcher), welche diese Farbe (Detectable) enthält.	
	<b>color</b> TransformedColor, <b>required</b>	Eine Farbe, welche durch Koordinaten im Farbraum angegeben wird. Die Array-Anzeichen des Besitzes des values entsprechen der Reihenfolge des Besitzes des color.space.axes des aktuellen Erkennungsprofils.	
		color	
		<b>values</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	Ort in einem Farbraum
	<b>representations</b> ColorRepresentations, optional, read-only	Vorberechnete visuelle Darstellung einer geeigneten Farbe zur Wiedergabe	
		representations	

		<b>RGB</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	RGB-Farrray, das die Achsen r, g und b in dieser Reihenfolge darstellt. Werte sind Gleitkommazahlen zwischen 0 und 1.
--	--	--	---

**Beispiele**

```
{
  "uuid": "9f968e8a-ad9c-45ce-9beb-a55011856a99",
  "alias": 2,
  "matcher_id": "1c7e9725-8753-4b6c-a0b7-a71d7e915cb5",
  "color": {
    "values": [
      0.476731,
      0.381263,
      0.128475
    ]
  },
  "representations": {
    "RGB": [
      0.396114,
      0.479113,
      0.552308
    ]
  }
}
```

**Antwort**

Code	Body	JSON-Objekt		
200 400	Properties ( <a href="#">object</a> )			
	<b>data</b> <a href="#">ColorDetectable</a> , <b>required</b>	Eine Farbe (Detectable) vertitt die numerische Position in einem Farbraum. Sie ist verbunden mit einer Farbgruppe (Matcher).		
		ColorDetectable		
		<b>uuid</b> <a href="#">UUID</a> (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
		<b>alias</b> <a href="#">Alias</a> (integer), <b>required</b> , read-only	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.	
		<b>matcher_id</b> <a href="#">UUID</a> (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	Referenz zu der Farbgruppe (Matcher), welche diese Farbe (Detectable) enthält.	

Code	Body	JSON-Objekt		
		<b>color</b> TransformedColor, required	Eine Farbe, welche durch Koordinaten im Farbraum angegeben wird. Die Array-Anzeichen des Besitzes des values entsprechen der Reihenfolge des Besitzes des color-space.axes des aktuellen Erkennungsprofils.	
			color	
			<b>values</b> Array of number, minimum items: 3, maximum items: 3, required	Ort in einem Farbraum
		<b>representations</b> ColorRepresentations, optional, read-only	Vorberechnete visuelle Darstellung einer geeigneten Farbe zur Wiedergabe	
			representations	
			<b>RGB</b> Array of number, minimum items: 3, maximum items: 3, required	RGB-Farbararray, das die Achsen r, g und b in dieser Reihenfolge darstellt. Werte sind Gleitkommazahlen zwischen 0 und 1.
	<b>errors</b> Array of Error, required	Error[]		
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode	
		<b>Mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte	
		<b>Message</b> String, optional	Visuell lesbare Fehlerbeschreibung	
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.validation.collection_size_exceeded			

### 1.2.2.12 Farbe löschen (ColorDetectables)

**DELETE** / sensor / detectables / {itemId}

Lösche eine einzige Farbe.

Anfrage

<b>Path Variables</b>
<b>itemId</b> String, required

Antwort

<b>Code</b>	
204	Die leere Antwort weist auf Erfolg hin
	<b>Könnte folgende Fehlercodes ausgeben</b> LCOL.samples.unavailable

### 1.2.2.13 Farbe ändern (ColorDetectable)

**PUT** / sensor / detectables / {itemId}

Ändert eine einzige Farbe (ColorDetectable).

Anfrage

Path Variables			
<b>itemId</b> String, <b>required</b>			
<b>Body</b>	<b>JSON-Objekt</b>		
Properties (colorDetectable)	<b>uuid</b> UUID (string), pattern: <code>^[a- €0-9-]+\$, <b>required</b>, read- only</code>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
	<b>alias</b> Alias (integer), <b>required</b> , read- only	Ein numerischer Wert, wel- cher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alterna- tive genutzt werden um die Gegenstände in URLs und an- deren Protokollen wie Modbus oder Schnittstellen anzuspre- chen.	
	<b>matcher_id</b> UUID (string), pattern: <code>^[a-f0- 9-]+\$, <b>required</b>, read-only</code>	Referenz zu der Farbgruppe (Matcher), welche diese Farbe (Detectable) enthält.	
	<b>color</b> TransformedColor, <b>required</b>	Eine Farbe, welche durch Ko- ordinaten im Farbraum ange- geben wird. Die Array-Anzei- chen des Besitzes des <code>va- lues</code> entsprechen der Reihen- folge des Besitzes des <code>co- lorspace.axes</code> des aktuel- len Erkennungsprofils.	
		<b>color</b>	
		<b>values</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	Ort in einem Farbraum
	<b>representations</b> ColorRepresentations, op- tional, read-only	Vorberechnete visuelle Darstel- lung einer geeigneten Farbe zur Wiedergabe	
		<b>representations</b>	
		<b>RGB</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	RGB-Farrray, das die Ach- sen r, g und b in dieser Rei- henfolge darstellt. Werte sind Gleitkommazahlen zwischen 0 und 1.

### Beispiele

```
{
  "uuid": "9f968e8a-ad9c-45ce-9beb-a55011856a99",
  "alias": 2,
  "matcher_id": "1c7e9725-8753-4b6c-a0b7-a71d7e915cb5",
  "color": {
    "values": [
      0.476731,
      0.381263,
      0.128475
    ]
  }
}
```

colorSENSOR CFO

```

},
"representations": {
  "RGB": [
    0.396114,
    0.479113,
    0.552308
  ]
}
}
}

```

Antwort

Code	Body	JSON-Objekt		
200 400 404	Properties ( <a href="#">object</a> )			
	<b>data</b> <a href="#">ColorDetectable</a> , <b>required</b>	Eine Farbe (Detectable) vertitt die numerische Position in einem Farbraum. Sie ist verbunden mit einer Farbgruppe (Matcher).		
		<a href="#">ColorDetectable</a>		
		<b>uuid</b> <a href="#">UUID</a> (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
		<b>alias</b> <a href="#">Alias</a> (integer), <b>required</b> , read-only	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.	
		<b>matcher_id</b> <a href="#">UUID</a> (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	Referenz zu der Farbgruppe (Matcher), welche diese Farbe (Detectable) enthält.	
		<b>color</b> <a href="#">TransformedColor</a> , <b>required</b>	Eine Farbe, welche durch Koordinaten im Farbraum angegeben wird. Die Array-Anzeichen des Besitzes des <code>values</code> entsprechen der Reihenfolge des Besitzes des <code>color-space.axes</code> des aktuellen Erkennungsprofils.	
			<code>color</code>	
			<b>values</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	Ort in einem Farbraum
		<b>representations</b> <a href="#">ColorRepresentations</a> , optional, read-only	Vorberechnete visuelle Darstellung einer	

Code	Body	JSON-Objekt		
			geeigneten Farbe zur Wiedergabe	
			representations	
			<b>RGB</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	RGB-Farrray, das die Achsen r, g und b in dieser Reihenfolge darstellt. Werte sind Gleitkommazahlen zwischen 0 und 1.
	<b>errors</b> Array of Error, <b>required</b>	Error[]		
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode	
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte	
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung	
	<b>Könnte folgende Fehlercodes ausgeben</b> LCOL.samples.unavailable			

### 1.2.2.14 Farbe auslesen (ColorDecetables)

**GET** / sensor / decetables / {itemId}

Gibt eine einzige Farbe (ColorDetectable) aus.

Anfrage

Path Variables
<b>itemId</b> String, <b>required</b>

Antwort

Code	Body	JSON-Objekt		
200	Properties ( <a href="#">object</a> )			
	<b>data</b> <a href="#">ColorDetectable</a> , <b>required</b>	Eine Farbe (Detectable) vertitt die numerische Position in einem Farbraum. Sie ist verbunden mit einer Farbgruppe (Matcher).		
		ColorDetectable		
		<b>uuid</b> <a href="#">UUID</a> (string), pattern: ^[a-f0-9-]+\$, <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
		<b>alias</b> <a href="#">Alias</a> (integer), <b>required</b> , read-only	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs	

Code	Body	JSON-Objekt		
			und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.	
		<b>matcher_id</b> UUID (string), pattern: ^ [a-f0-9-]+\$, <b>required</b> , read-only	Referenz zu der Farbgruppe (Matcher), welche diese Farbe (Detectable) enthält.	
		<b>color</b> TransformedColor, <b>required</b>	Eine Farbe, welche durch Koordinaten im Farbraum angegeben wird. Die Array-Anzeichen des Besitzes des values entsprechen der Reihenfolge des Besitzes des color-space.axes des aktuellen Erkennungsprofils.	
			color	
			<b>values</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	Ort in einem Farbraum
		<b>representations</b> ColorRepresentations, optional, read-only	Vorberechnete visuelle Darstellung einer geeigneten Farbe zur Wiedergabe	
			representations	
			<b>RGB</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	RGB-Farbararray, das die Achsen r, g und b in dieser Reihenfolge darstellt. Werte sind Gleitkommazahlen zwischen 0 und 1.
	<b>errors</b> Array of Error, <b>required</b>	Error[]		
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode	
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte	
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung	
	<b>Könnte folgende Fehlercodes ausgeben</b> LCOL.samples.unavailable			

### 1.2.2.15 Wechsel aktuelles Erkennungsprofil

**PUT** / sensor / detection-profiles

Nur eins der verfügbaren Erkennungsprofile ist aktiv zu einer gegebenen Zeit. Schreibe eine neue Erkennungsprofil-ID in das `current_profile_id` Feld, um das aktuell benutzte Profil zu ändern.

Anfrage

Body	JSON-Objekt
------	-------------

Examples Example	a014e415-0fec-4734-ac3f-30da0a5f3899
---------------------	--------------------------------------

Antwort

Code	Body	JSON-Objekt	
200	Properties (object)		
	<b>data</b> <u>CurrentDetectionProfileID</u> (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	Der Sensor kann mehrere Erkennungsprofile speichern, aber er kann nur jeweils eins anwenden. Das Feld <code>current_profile_id</code> beinhaltet die UUID des momentan vom Sensor benutzten Erkennungsprofils für seinen Betrieb. Es erlaubt den API-Verknüpfungsendpunkt <code>/api/sensor(detection-profile/current)</code> zu benutzen, anstatt ein Erkennungsprofil durch dessen UUID festzulegen.	
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung

1.2.2.16 Erstelle ein Erkennungsprofil

**POST** / sensor / detection-profiles

Erstelle ein neues Erkennungsprofil.

Alle unterstützten Datenattribute im Body der Anfrage sind optional.

Body	JSON-Objekt		
Properties (DetectionProfile)	<b>uuid</b> <u>UUID</u> (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
	<b>alias</b> <u>Alias</u> (integer), <b>required</b> , read-only	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.	
	<b>name</b> String, <b>required</b>	Visuell lesbarer Name des Erkennungsprofils.	
	<b>colorspace</b> Colorspace, <b>required</b>	Ein Farbraum beschreibt die numerische Umwandlung von Farben unter bestimmten Bedingungen. Verschiedene standardisierte Farbräume sind passend für verschiedene Erkennungsaufgaben.	
		colorspace	
		<b>name</b> String, <b>required</b>	
		<b>space_id</b> ColorspaceID, <b>required</b>	Eindeutiger Name eines Farbraums

Body	JSON-Objekt			
		<b>axes</b> Array of ColorspaceAxis, minimum items: 3, maximum items: 3, <b>required</b>	ColorspaceAxis[]	
			<b>id</b> String, <b>required</b>	Eindeutiger Name
			<b>label</b> String, <b>required</b>	Visuell lesbare Name
			<b>minimum</b> Number, <b>required</b>	Niedrigster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
			<b>maximum</b> Number, <b>required</b>	Höchster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
	<b>non_matching_output</b> WantedSwitchingOutputsState, <b>required</b>	Dieser Zustand der Schaltausgänge wird angewandt, wenn die aktuelle Stichprobenfarbe zu keiner gespeicherten Farbgruppe (Matcher) gehört.		
		non_matching_output		
		<b>uuid</b> UUID (string), pattern: ^[a-f0-9-]+\$, <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
		<b>states</b> Array of any of boolean or null, <b>required</b>	Liste von True/False/Null Werten beschreibt den gewollten Zustand des Schaltausgangs	
	<b>non_matching_hold_time</b> HoldTime (number), maximum 3153600000, <b>required</b>	Minimale Dauer (in Sekunden) der Anwendung des <i>non_matching_output</i> Zustands am Schaltausgang des Sensors. Diese Verlängerung eines potentiellen <i>nicht passenden</i> Falls könnte nützlich sein, wenn die Bearbeitungszeit eines verbundenen Aktors den Stichprobenzeitraum des Sensors überschreitet.		
	<b>compensation_settings</b> CompensationSettings, <b>required</b>	Die Kompensationseinstellungen eines Erkennungsprofils beschreiben die Konfiguration eines internen Sensorkomponenten, bezogen auf die Stabilisations- und Kompensationsalgorithmen. Diese Werte können mittels Erteilung einer <b>POST</b> Anfrage entgegen <code>/api/sensor/detection-profiles/current/autogain</code> bestimmt werden. Das Ergebnis ist ein passendes Set aus Kompensationseinstellungen für diesen Sensor unter den aktuellen Umständen. Der Inhalt dieses Datenobjekts ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).		
		compensation_settings		

Body	JSON-Objekt			
	<b>sampling_settings</b> SamplingSettings, required	<p>Stichprobeneinstellungen beschreiben alle Details der Stichprobenentnahme.</p> <p>Ihre Attribute können abgefragt und untersucht werden (z.B. um die aktuelle Abtaste abzurufen).</p> <p>Die meisten Werte die in den Stichprobeneinstellungen gespeichert sind sollten nicht direkt geändert werden. Der verbundene API Endpunkt <code>/api/sensor/detection-profiles/current/autogain</code> sollte stattdessen benutzt werden.</p> <p>Das einzige veränderbare Attribut innerhalb der Stichprobeneinstellungen ist der Durchschnittswert. Es ist sicher ihn zu ändern, obwohl die berechneten Standardwerte während eines Autogain-Betriebs optimal für die meisten Erkennungsaufgaben sein sollten.</p>		
		sampling_settings		
		<b>led_intensity</b> Number, minimum: 0, maximum: 1, required	Relative Intensität des Emitters während der Leuchtphase	
		<b>base_sample_rate</b> SampleRate (number), minimum: 0.01, required	<p>Die Basisabtaste bestimmt die Dauer der Abtastperiode.</p> <p>Nach jeder Abtastperiode, werden die erfassten Daten verarbeitet und ein neues Erfassungsergebnis wird berechnet (z.B. die passendste Farbgruppe (Matcher) für das gegebene Beispiel). Dies könnte den Zustand der Schaltausgänge beeinträchtigen oder konfigurierte Aktionen auslösen. Somit definiert die Basisabtaste die Maximumrate der Änderungen für die Schaltausgänge.</p> <p>Siehe ebenso die effektive Abtaste.</p>	
		<b>effective_sample_rate</b> SampleRate (number), minimum: 0.01, required	<p>Die effektive Abtaste ist ein numerisches Produkt der Basisabtaste und der Anzahl der Durchschnitte.</p> <p>Sie bestimmt die Minimumdauer die ein Ziel benötigt um abgetastet zu werden um dessen visuellen Auftritt korrekt zu bestimmen.</p> <p>Mit dem Einstellen der Standardwerte auf eins, ist dieser Wert gleich zu der Basisabtaste.</p>	
		<b>minimum_wanted_sample_rate</b> SampleRate (number), minimum: 0.01, required	<p>Dieser informative Wert repräsentiert die Abtaste, welche während des aktuellsten Autogain-Vorgangs angefragt wurde.</p> <p>Die effektive Abtaste könnte von der gewollten Abtaste abweichen, falls die angefragte Abtaste aufgrund der Grenzen des Sensors nicht erreichbar war (z.B. Überschreiten der unterstützten Abtaste) oder aufgrund der Umwelt (z.B. nicht genug Licht, sodass eine langsamere Amplifikation mit höherer Verstärkung nötig war).</p>	

Body	JSON-Objekt		
		<b>sample_light_phase</b> Boolean, <b>required</b>	Definiert ob der Sensor die internen Emitter zum Abtasten periodisch aktiviert soll
		<b>sample_dark_phase</b> Boolean, <b>required</b>	Definiert ob der Sensor die internen Emitter zum Abtasten periodisch deaktiviert soll.
		<b>averages</b> AverageSampleCount (integer), minimum: 1, <b>required</b>	Anzahl vorheriger Stichproben die für jedes Stichprobenergebnis gemittelt werden sollen. Ein rollierender Mittelungsalgorithmus wird an den Stichproben angewandt.
		<b>amplification</b> AmplificationLevel (integer), <b>required</b>	Der Amplifikationslevel gibt die interne Konfiguration eines Verstärkers an. Der Wert ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).
	<b>white_reference</b> Array of number, <b>required</b>	Das Weißreferenzattribut wird benutzt um einen benutzerdefinierten Farbausgleich anzugeben. Sein Inhalt unterliegt dem internen Gebrauch. Somit sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte (z.B. <code>/api/sensor/detection-profiles/{itemId}/white-reference</code> ).	
	<b>normalization_constant</b> Array of number, <b>required</b>	Normierungskonstanten sind verbunden mit der Weißreferenz. Ihr Inhalt unterliegt dem internen Gebrauch. Somit sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte (z.B. <code>/api/sensor/detection-profiles/{itemId}/white-reference</code> ).	

## Beispiele

```
{
  "name": "#0",
  "uuid": "2475df8d-85f0-4208-ba60-dce6cb282a96",
  "alias": 1,
  "non_matching_hold_time": 0,
  "colorspace": {
    "name": "L*a*b*",
    "axes": [
      {
        "id": "L",
        "label": "L*",
        "minimum": 0,
        "maximum": 100
      },
      {
        "id": "a",
        "label": "a*",
        "minimum": -500,
        "maximum": 500
      }
    ]
  }
}
```

```

        "id": "b",
        "label": "b*",
        "minimum": -200,
        "maximum": 200
    }
],
"space_id": "Lab"
},
"compensation_settings": {
    "monitor_integration": {
        "control": 0.32499998807907104,
        "references": [
            0.7283520102500916,
            0.7442666888237,
            0.7066696286201477
        ]
    },
    "use_calibration_samples": true
},
"normalization_constant": [
    237.4935277662995,
    242.62655153828055,
    587.8264132734112
],
"white_reference": [
    95.047,
    100,
    108.883
],
"non_matching_output": {
    "uuid": "3f26aff4-8650-42a0-b319-51776c443fbc",
    "states": [
        true,
        true,
        true,
        true,
        true,
        true,
        true,
        true
    ]
},
"sampling_settings": {
    "led_intensity": 1,
    "amplification": 1,
    "sample_light_phase": true,
    "minimum_wanted_sample_rate": 1000,
    "averages": 1,
    "base_sample_rate": 1000,
    "sample_dark_phase": true,
    "effective_sample_rate": 1000
}
}

```

**Antwort**

Code	Body	JSON-Objekt			
200 400	Properties (object)				
	<b>data</b> <a href="#">DetectionProfile</a> , <b>required</b>	Ein Erkennungsprofil erkennt ein komplettes			

Code	Body	JSON-Objekt			
		<p>Set aus Sensoreinstellungen für eine gegebene Erkennungsaufgabe.                      Mehrere Profile können gespeichert werden, um zwischen verschiedenen Erkennungsaufgaben zu wechseln oder für die inkrementelle Entwicklung eines verfeinerten Profils.                      Manche Attribute eines Erkennungsprofils legen interne Details des Sensors offen, welche indirekt durch andere Mittel bestimmt werden sollten. Diese Attribute werden nur oberflächlich beschrieben, da sie so behandelt werden sollen wie sie sind ohne ihre Werte oder Struktur zu ändern.</p>			
		DetectionProfile			
		<p><b>uuid</b>  <a href="#">UUID</a> (string), pattern: <code>^[a-f0-9-]+\$</code>, <b>required</b>, read-only</p>	<p>eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8</p>		
		<p><b>alias</b>  <a href="#">Alias</a> (integer), <b>required</b>, read-only</p>	<p>Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.</p>		
		<p><b>name</b>                      String, <b>required</b></p>	<p>Visuell lesbarer Name des Erkennungsprofils</p>		
		<p><b>colorspace</b>                      Colorspace, <b>required</b></p>	<p>Ein Farbraum beschreibt die numerische Umwandlung von Farben unter bestimmten Bedingungen. Verschiedene standardisierte Farbräume sind passend für verschiedene Erkennungsaufgaben.</p>		
			colorspace		
			<p><b>name</b>                      String, <b>required</b></p>		
			<p><b>space_id</b>                      ColorspaceID, <b>required</b></p>	Eindeutiger Name eines Farbraums	
			<p><b>axes</b>                      Array of ColorspaceAxis, minimum items: 3, maximum items: 3, <b>required</b></p>	ColorspaceAxis[]	
				<p><b>id</b>                      String, <b>required</b></p>	Eindeutiger Name
				<p><b>label</b>                      String, <b>required</b></p>	Visuell lesbarer Name

Code	Body	JSON-Objekt			
				<b>minimum</b> Number, <b>required</b>	Niedrigster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
				<b>maximum</b> Number, <b>required</b>	Höchster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
		<b>non_matching_output</b> WantedSwitchingOutputsState, <b>required</b>	Dieser Zustand der Schaltausgänge wird angewandt, wenn die aktuelle Stichprobenfarbe zu keiner gespeicherten Farbgruppe (Matcher) gehört.		
			non_matching_output		
			<b>uuid</b> UUID (string), pattern: ^[a-f0-9-]+\$, <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
			<b>states</b> Array of any of boolean or null, <b>required</b>	Liste von True/False/Null Werten beschreibt den gewollten Zustand des Schaltausgangs	
		<b>non_matching_hold_time</b> HoldTime (number), maximum 3153600000, <b>required</b>	Minimale Dauer (in Sekunden) der Anwendung des <i>non_matching_output</i> Zustands am Schaltausgang des Sensors. Diese Verlängerung eines potentiellen <i>nicht passenden</i> Falls könnte nützlich sein, wenn die Bearbeitungszeit eines verbundenen Aktors den Stichprobenzeitraum des Sensors überschreitet.		
		<b>compensation_settings</b> CompensationSettings, <b>required</b>	Die Kompensationseinstellungen eines Erkennungsprofils beschreiben die Konfiguration eines internen Sensorkomponenten, bezogen auf die Stabilisations- und Kompensationsalgorithmen.  Diese Werte können mittels Erteilung einer <b>POST</b> Anfrage entgegen <code>/api/sensor/detection-profiles/current/autogain</code> bestimmt werden. Das Ergebnis ist ein passendes Set aus Kompensationseinstellungen für diesen Sensor unter den aktuellen Umständen.  Der Inhalt dieses Datenobjekts ist nicht dazu gedacht		

Code	Body	JSON-Objekt			
			von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).		
			compensation_settings		
		<b>sampling_settings</b> SamplingSettings, required	<p>Stichprobeneinstellungen beschreiben alle Details der Stichprobenentnahme. Ihre Attribute können abgefragt und untersucht werden (z.B. um die aktuelle Abtastrate abzurufen).</p> <p>Die meisten Werte die in den Stichprobeneinstellungen gespeichert sind sollten nicht direkt geändert werden. Der verbundene API Endpunkt /api/sensor/detection-profiles/current/autogain sollte stattdessen benutzt werden.</p> <p>Das einzige veränderbare Attribut innerhalb der Stichprobeneinstellungen ist der Durchschnittswert. Es ist sicher ihn zu ändern, obwohl die berechneten Standardwerte während eines Autogain-Betriebs optimal für die meisten Erkennungsaufgaben sein sollten.</p>		
			sampling_settings		
			<b>led_intensity</b> Number, minimum: 0, maximum: 1, required	Relative Intensität des Emitters während der Leuchtphase	
			<b>base_sample_rate</b> SampleRate (number), minimum: 0.01, required	Die Basisabtastrate bestimmt die Dauer der Abtastperiode. Nach jeder Abtastperiode, werden die erfassten Daten verarbeitet und ein neues Erfassungsergebnis wird berechnet (z.B. die passendste Farbgruppe (Matcher) für das gegebene Beispiel). Dies könnte den Zustand der Schaltausgänge beeinträchtigen oder konfigurierte Aktionen auslösen. Somit definiert die Basisabtastrate die Maximumrate der Änderungen für die Schaltausgänge. Siehe ebenso die effektive Abtastrate.	

Code	Body	JSON-Objekt		
			<b>effective_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Die effektive Abtastrate ist ein numerisches Produkt der Basisabtastrate und der Anzahl der Durchschnitte. Sie bestimmt die Minimumdauer die ein Ziel benötigt um abgetastet zu werden um dessen visuellen Auftritt korrekt zu bestimmen. Mit dem Einstellen der Standardwerte auf eins, ist dieser Wert gleich zu der Basisabtastrate.
			<b>minimum_wanted_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Dieser informative Wert repräsentiert die Abtastrate, welche während des aktuellsten Autogain-Vorgangs angefragt wurde. Die effektive Abtastrate könnte von der gewollten Abtastrate abweichen, falls die angefragte Abtastrate aufgrund der Grenzen des Sensors nicht erreichbar war (z.B. Überschreiten der unterstützten Abtastrate) oder aufgrund der Umwelt (z.B. nicht genug Licht, sodass eine langsamere Amplifikation mit höherer Verstärkung nötig war).
			<b>sample_light_phase</b> Boolean, <b>required</b>	Definiert ob der Sensor die internen Emitter zum Abtasten periodisch aktivierten soll
			<b>sample_dark_phase</b> Boolean, <b>required</b>	Definiert ob der Sensor die internen Emitter zum Abtasten periodisch deaktivieren soll.
			<b>averages</b> AverageSampleCount (integer), minimum: 1, <b>required</b>	Anzahl vorheriger Stichproben die für jedes Stichprobenergebnis gemittelt werden sollen. Ein rollierender Mittelungsalgorithmus wird an den Stichproben angewandt.

Code	Body	JSON-Objekt				
			<b>amplification</b> AmplificationLevel (integer), <b>required</b>	Der Amplifikationslevel gibt die interne Konfiguration eines Verstärkers an. Der Wert ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).		
		<b>white_reference</b> Array of number, <b>required</b>	Das Weißreferenzattribut wird benutzt um einen benutzerdefinierten Farbausgleich anzugeben. Sein Inhalt unterliegt dem internen Gebrauch. Somit sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte (z.B. /api/sensor/detection-profiles/{itemId}/white-reference).			
		<b>normalization_constant</b> Array of number, <b>required</b>	Normierungskonstanten sind verbunden mit der Weißreferenz. Ihr Inhalt unterliegt dem internen Gebrauch. Somit sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte (z.B. /api/sensor/detection-profiles/{itemId}/white-reference).			
	<b>errors</b> Array of Error, <b>required</b>	Error[]				
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode			
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte			
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung			
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.validation.collection_size_exceeded					

### 1.2.2.17 Entferne mehrere oder alle Erkennungsprofile

**DELETE** / sensor / detection-profiles

Entferne eine Auswahl an Erkennungsprofilen. Entweder basierend auf einem gegebenem Filterargument (falls unterstützt für diese Sammlung) oder entferne alle Erkennungsprofile aus der Sammlung.

Alle Lösch-Anfragen resultieren in einer leeren Erfolgsrückmeldung (204). Dies ist sogar gültig für eine ungefilterte **DELETE**-Anfrage entgegen einer leeren Sammlung oder für eine gefilterte **DELETE**-Anfrage entgegen einer Sammlung ohne Erkennungsprofile die dem Filter entsprechen.

Antwort

Code	Body	JSON-Objekt	
200 204	Properties (object)		
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>data</b> Object, <b>required</b>	data	
		<b>current_profile_id</b> <u>CurrentDetectionProfileID</u> (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	Der Sensor kann mehrere Erkennungsprofile speichern, aber er kann nur jeweils eins anwenden. Das Feld <code>current_profile_id</code> beinhaltet die UUID des momentan vom Sensor benutzten Erkennungsprofils für seinen Betrieb. Es erlaubt den API-Verknüpfungsendpunkt <code>/api/sensor(detection-profile/current)</code> zu benutzen, anstatt ein Erkennungsprofil durch dessen UUID festzulegen.

1.2.2.18 Erkennungsprofile auslesen

**GET** / sensor / detection-profiles

Liest eine Liste verfügbarer Erkennungsprofile aus.

Antwort

Code	Body	JSON-Objekt			
200	Properties (object)				
	<b>data</b> Object, <b>required</b>	<b>data</b>			
		<b>detection-profiles</b> Array of Detection-Profile, <b>required</b>	DetectionProfile[]		
			<b>uuid</b> <u>UUID</u> (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
			<b>alias</b> <u>Alias</u> (integer), <b>required</b> , read-only	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.	
			<b>name</b> String, <b>required</b>	Visuell lesbarer Name des Erkennungsprofils	
			<b>colorspace</b> Colorspace, <b>required</b>	Ein Farbraum beschreibt die numerische Umwandlung	

Code	Body	JSON-Objekt				
				von Farben unter bestimmten Bedingungen. Verschiedene standardisierte Farbräume sind passend für verschiedene Erkennungsaufgaben.		
				colorspace		
				<b>name</b> String, <b>required</b>		
				<b>space_id</b> ColorspaceID, <b>required</b>	Eindeutiger Name eines Farbraums	
				<b>axes</b> Array of ColorspaceAxis, minimum items: 3, maximum items: 3, <b>required</b>	ColorspaceAxis[]	
					<b>id</b> String, <b>required</b>	Eindeutiger Name
					<b>label</b> String, <b>required</b>	Visuell lesbarer Name
					<b>minimum</b> Number, <b>required</b>	Niedrigster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
					<b>maximum</b> Number, <b>required</b>	Höchster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
			<b>non_matching_output</b> WantedSwitchingOutputsState, <b>required</b>	Dieser Zustand der Schaltausgänge wird angewandt, wenn die aktuelle Stichprobenfarbe zu keiner gespeicherten Farbgruppe (Matcher) gehört.		
				non_matching_output		
				<b>uuid</b> UUID (string), pattern: ^[a-f0-9-]+\$, <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
				<b>states</b> Array of any of boolean or null, <b>required</b>	Liste von True/False/Null Werten beschreibt den gewollten Zustand des Schaltausgangs	
			<b>non_matching_hold_time</b> HoldTime (number), maximum 3153600000, <b>required</b>	Minimale Dauer (in Sekunden) der Anwendung des <i>non_matching_output</i> Zustands am Schaltausgang des Sensors. Diese Verlängerung eines potentiellen <i>nicht passenden</i> Falls könnte nützlich sein, wenn die Bearbeitungszeit eines		

Code	Body	JSON-Objekt			
				verbundenen Aktors den Stichprobenzeitraum des Sensors überschreitet.	
			<b>compensation_settings</b> CompensationSettings, required	<p>Die Kompensationseinstellungen eines Erkennungsprofils beschreiben die Konfiguration eines internen Sensorkomponenten, bezogen auf die Stabilisations- und Kompensationsalgorithmen.</p> <p>Diese Werte können mittels Erteilung einer <b>POST</b> Anfrage entgegen <code>/api/sensor/detection-profiles/current/autogain</code> bestimmt werden. Das Ergebnis ist ein passendes Set aus Kompensationseinstellungen für diesen Sensor unter den aktuellen Umständen.</p> <p>Der Inhalt dieses Datenobjekts ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).</p>	
				compensation_settings	
			<b>sampling_settings</b> SamplingSettings, required	<p>Stichprobeneinstellungen beschreiben alle Details der Stichprobenentnahme. Ihre Attribute können abgefragt und untersucht werden (z.B. um die aktuelle Abtastrate abzurufen).</p> <p>Die meisten Werte die in den Stichprobeneinstellungen gespeichert sind sollten nicht direkt geändert werden. Der verbundene API Endpunkt <code>/api/sensor/detection-profiles/current/autogain</code> sollte stattdessen benutzt werden.</p> <p>Das einzige veränderbare Attribut innerhalb der Stichprobeneinstellungen ist der Durchschnittswert. Es ist sicher ihn zu ändern, obwohl die berechneten Standardwerte während eines Autogain-Betriebs optimal für die meisten Erkennungsaufgaben sein sollten.</p>	
				sampling_settings	
				<b>led_intensity</b> Number, minimum: 0, maximum: 1, required	Relative Intensität des Emitters während der Leuchtphase
				<b>base_sample_rate</b> SampleRate (number), minimum: 0.01, required	Die Basisabtastrate bestimmt die Dauer der Abtastperiode. Nach jeder Abtastperiode, werden die erfassten

Code	Body	JSON-Objekt			
					Daten verarbeitet und ein neues Erfassungsergebnis wird berechnet (z.B. die passendste Farbgruppe (Matcher) für das gegebene Beispiel). Dies könnte den Zustand der Schaltausgänge beeinträchtigen oder konfigurierte Aktionen auslösen. Somit definiert die Basisabtastrate die Maximumrate der Änderungen für die Schaltausgänge. Siehe ebenso die effektive Abtastrate.
				<b>effective_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Die effektive Abtastrate ist ein numerisches Produkt der Basisabtastrate und der Anzahl der Durchschnitte. Sie bestimmt die Minimumdauer die ein Ziel benötigt um abgetastet zu werden um dessen visuellen Auftritt korrekt zu bestimmen. Mit dem Einstellen der Standardwerte auf eins, ist dieser Wert gleich zu der Basisabtastrate.
				<b>minimum_wanted_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Dieser informative Wert repräsentiert die Abtastrate, welche während des aktuellsten Autogain-Vorgangs angefragt wurde. Die effektive Abtastrate könnte von der gewollten Abtastrate abweichen, falls die angefragte Abtastrate aufgrund der Grenzen des Sensors nicht erreichbar war (z.B. Überschreiten der unterstützten Abtastrate) oder aufgrund der Umwelt (z.B. nicht genug Licht, sodass eine langsamere Amp-

Code	Body	JSON-Objekt				
						lifikation mit höherer Verstärkung nötig war).
				<b>sample_light_phase</b> Boolean, <b>required</b>		Definiert ob der Sensor die internen Emitter zum Abtasten periodisch aktivierten soll
				<b>sample_dark_phase</b> Boolean, <b>required</b>		Definiert ob der Sensor die internen Emitter zum Abtasten periodisch deaktivieren soll.
				<b>averages</b> AverageSampleCount (integer), minimum: 1, <b>required</b>		Anzahl vorheriger Stichproben die für jedes Stichprobenergebnis gemittelt werden sollen. Ein rollierender Mittelungsalgorithmus wird an den Stichproben angewandt.
				<b>amplification</b> AmplificationLevel (integer), <b>required</b>		Der Amplifikationslevel gibt die interne Konfiguration eines Verstärkers an. Der Wert ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).
			<b>white_reference</b> Array of number, <b>required</b>	Das Weißreferenzattribut wird benutzt um einen benutzerdefinierten Farbausgleich anzugeben. Sein Inhalt unterliegt dem internen Gebrauch. Somit sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte (z.B. /api/sensor/detection-profiles/{itemId}/white-reference).		
			<b>normalization_constant</b> Array of number, <b>required</b>	Normierungskonstanten sind verbunden mit der Weißreferenz. Ihr Inhalt unterliegt dem internen Gebrauch. Somit sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte (z.B. /api/sensor/detection-profiles/{itemId}/white-reference).		
		<b>current_profile_id</b>	Der Sensor kann mehrere Erkennungsprofile			

Code	Body	JSON-Objekt				
		<u>CurrentDetectionProfileID</u> (string), pattern: <code>^[a-f0-9-]+</code> , <b>required</b> , read-only	speichern, aber er kann nur jeweils eins anwenden. Das Feld <code>current_profile_id</code> beinhaltet die UUID des momentan vom Sensor benutzten Erkennungsprofils für seinen Betrieb. Es erlaubt den API-Verknüpfungsendpunkt <code>/api/sensor(detection-profile/current)</code> zu benutzen, anstatt ein Erkennungsprofil durch dessen UUID festzulegen.			
	<b>errors</b> Array of Error, <b>required</b>	Error[]				
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode			
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte			
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung			

### 1.2.2.19 Erkennungsprofil löschen

**DELETE** / sensor / detection-profiles / {itemId}

Löscht ein einziges Erkennungsprofil.

Anfrage

Path Variables
<b>itemId</b> String, <b>required</b>

Antwort

Code	
204	Die leere Antwort weist auf Erfolg hin
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item

### 1.2.2.20 Erkennungsprofil ändern

**PUT** / sensor / detection-profiles / {itemId}

Ändert ein einziges Erkennungsprofil.

Anfrage

Path Variables
<b>itemId</b> String, <b>required</b>

Body	JSON-Objekt			
Properties (DetectionProfile)	<b>uuid</b> UUID (string), pattern: <code>^[a-f0-9-]+</code> , <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8		

Body	JSON-Objekt			
	<b>alias</b> <i>Alias</i> (integer), <b>required</b> , read-only	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.		
	<b>name</b> String, <b>required</b>	Visuell lesbarer Name des Erkennungsprofils		
	<b>colorspace</b> Colorspace, <b>required</b>	Ein Farbraum beschreibt die numerische Umwandlung von Farben unter bestimmten Bedingungen. Verschiedene standardisierte Farbräume sind passend für verschiedene Erkennungsaufgaben.		
		colorspace		
		<b>name</b> String, <b>required</b>		
		<b>space_id</b> ColorspaceID, <b>required</b>	Eindeutiger Name eines Farbraums	
		<b>axes</b> Array of ColorspaceAxis, minimum items: 3, maximum items: 3, <b>required</b>	ColorspaceAxis[]	
			<b>id</b> String, <b>required</b>	Eindeutiger Name
			<b>label</b> String, <b>required</b>	Visuell lesbarer Name
			<b>minimum</b> Number, <b>required</b>	Niedrigster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
			<b>maximum</b> Number, <b>required</b>	Höchster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
	<b>non_matching_output</b> WantedSwitchingOutputsState, <b>required</b>	Dieser Zustand der Schaltausgänge wird angewandt, wenn die aktuelle Stichprobenfarbe zu keiner gespeicherten Farbgruppe (Matcher) gehört.		
		non_matching_output		
		<b>uuid</b> UUID (string), pattern: ^[a-f0-9-]+\$, <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
		<b>states</b> Array of any of boolean or null, <b>required</b>	Liste von True/False/Null Werten beschreibt den gewollten Zustand des Schaltausgangs	
	<b>non_matching_hold_time</b> HoldTime (number), maximum 3153600000, <b>required</b>	Minimale Dauer (in Sekunden) der Anwendung des <i>non_matching_output</i> Zustands am Schaltausgang des Sensors. Diese Verlängerung eines potentiellen <i>nicht passenden</i> Falls könnte nützlich sein, wenn die		

Body	JSON-Objekt		
		Bearbeitungszeit eines verbundenen Aktors den Stichprobenzeitraum des Sensors überschreitet.	
	<b>compensation_settings</b> CompensationSettings, <b>required</b>	Die Kompensationseinstellungen eines Erkennungsprofils beschreiben die Konfiguration eines internen Sensorkomponenten, bezogen auf die Stabilisations- und Kompensationsalgorithmen. Diese Werte können mittels Erteilung einer <b>POST</b> Anfrage entgegen <code>/api/sensor/detection-profiles/current/autogain</code> bestimmt werden. Das Ergebnis ist ein passendes Set aus Kompensationseinstellungen für diesen Sensor unter den aktuellen Umständen. Der Inhalt dieses Datenobjekts ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).	
		compensation_settings	
	<b>sampling_settings</b> SamplingSettings, <b>required</b>	Stichprobeneinstellungen beschreiben alle Details der Stichprobenentnahme. Ihre Attribute können abgefragt und untersucht werden (z.B. um die aktuelle Abtastrate abzurufen). Die meisten Werte die in den Stichprobeneinstellungen gespeichert sind sollten nicht direkt geändert werden. Der verbundene API Endpunkt <code>/api/sensor/detection-profiles/current/autogain</code> sollte stattdessen benutzt werden. Das einzige veränderbare Attribut innerhalb der Stichprobeneinstellungen ist der Durchschnittswert. Es ist sicher ihn zu ändern, obwohl die berechneten Standardwerte während eines Autogain-Betriebs optimal für die meisten Erkennungsaufgaben sein sollten.	
		sampling_settings	
		<b>led_intensity</b> Number, minimum: 0, maximum: 1, <b>required</b>	Relative Intensität des Emitters während der Leuchtphase
		<b>base_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Die Basisabtastrate bestimmt die Dauer der Abtastperiode. Nach jeder Abtastperiode, werden die erfassten Daten verarbeitet und ein neues Erfassungsergebnis wird berechnet (z.B. die passendste Farbgruppe (Matcher) für das gegebene Beispiel). Dies könnte den Zustand der Schaltausgänge beeinträchtigen oder konfigurierte Aktionen auslösen. Somit definiert die Basisabtastrate die Maximumrate

Body	JSON-Objekt		
			der Änderungen für die Schaltausgänge. Siehe ebenso die effektive Abtastrate.
		<b>effective_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Die effektive Abtastrate ist ein numerisches Produkt der Basisabtastrate und der Anzahl der Durchschnitte. Sie bestimmt die Minimumdauer die ein Ziel benötigt um abgetastet zu werden um dessen visuellen Auftritt korrekt zu bestimmen. Mit dem Einstellen der Standardwerte auf eins, ist dieser Wert gleich zu der Basisabtastrate.
		<b>minimum_wanted_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Dieser informative Wert repräsentiert die Abtastrate, welche während des aktuellsten Autogain-Vorgangs angefragt wurde. Die effektive Abtastrate könnte von der gewollten Abtastrate abweichen, falls die angefragte Abtastrate aufgrund der Grenzen des Sensors nicht erreichbar war (z.B. Überschreiten der unterstützten Abtastrate) oder aufgrund der Umwelt (z.B. nicht genug Licht, sodass eine langsamere Amplifikation mit höherer Verstärkung nötig war).
		<b>sample_light_phase</b> Boolean, <b>required</b>	Definiert ob der Sensor die internen Emitter zum Abtasten periodisch aktivierten soll
		<b>sample_dark_phase</b> Boolean, <b>required</b>	Definiert ob der Sensor die internen Emitter zum Abtasten periodisch deaktivieren soll.
		<b>averages</b> AverageSampleCount (integer), minimum: 1, <b>required</b>	Anzahl vorheriger Stichproben die für jedes Stichprobenergebnis gemittelt werden sollen. Ein rollierender Mittelungsalgorithmus wird an den Stichproben angewandt.
		<b>amplification</b> AmplificationLevel (integer), <b>required</b>	Der Amplifikationslevel gibt die interne Konfiguration eines Verstärkers an. Der Wert ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden

Body	JSON-Objekt			
				wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).
	<b>white_reference</b> Array of number, <b>required</b>	Das Weißreferenzattribut wird benutzt um einen benutzerdefinierten Farbausgleich anzugeben. Sein Inhalt unterliegt dem internen Gebrauch. Somit sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte (z.B. /api/sensor/detection-profiles/{itemId}/white-reference).		
	<b>normalization_constant</b> Array of number, <b>required</b>	Normierungskonstanten sind verbunden mit der Weißreferenz. Ihr Inhalt unterliegt dem internen Gebrauch. Somit sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte (z.B. /api/sensor/detection-profiles/{itemId}/white-reference).		

**Beispiele**

```
{
  "name": "#0",
  "uuid": "2475df8d-85f0-4208-ba60-dce6cb282a96",
  "alias": 1,
  "non_matching_hold_time": 0,
  "colorspace": {
    "name": "L*a*b*",
    "axes": [
      {
        "id": "L",
        "label": "L*",
        "minimum": 0,
        "maximum": 100
      },
      {
        "id": "a",
        "label": "a*",
        "minimum": -500,
        "maximum": 500
      },
      {
        "id": "b",
        "label": "b*",
        "minimum": -200,
        "maximum": 200
      }
    ]
  },
  "space_id": "Lab"
},
"compensation_settings": {
  "monitor_integration": {
    "control": 0.32499998807907104,
    "references": [
      0.7283520102500916,
      0.7442666888237,
      0.7066696286201477
    ]
  }
}
```

```

    ]
  },
  "use_calibration_samples": true
},
"normalization_constant": [
  237.4935277662995,
  242.62655153828055,
  587.8264132734112
],
"white_reference": [
  95.047,
  100,
  108.883
],
"non_matching_output": {
  "uuid": "3f26aff4-8650-42a0-b319-51776c443fbc",
  "states": [
    true,
    true,
    true,
    true,
    true,
    true,
    true,
    true
  ]
}
},
"sampling_settings": {
  "led_intensity": 1,
  "amplification": 1,
  "sample_light_phase": true,
  "minimum_wanted_sample_rate": 1000,
  "averages": 1,
  "base_sample_rate": 1000,
  "sample_dark_phase": true,
  "effective_sample_rate": 1000
}
}

```

Antwort

Code	Body	JSON-Objekt			
200 400 404	Properties (object)				
	<b>data</b> <a href="#">DetectionProfile</a> , <b>required</b>	Ein Erkennungsprofil erkennt ein komplettes Set aus Sensoreinstellungen für eine gegebene Erkennungsaufgabe. Mehrere Profile können gespeichert werden, um zwischen verschiedenen Erkennungsaufgaben zu wechseln oder für die inkrementelle Entwicklung eines verfeinerten Profils. Manche Attribute eines Erkennungsprofils legen interne Details des Sensors offen, welche indirekt durch andere Mittel bestimmt werden sollten. Diese Attribute werden nur oberflächlich beschrieben, da sie so behandelt werden sollen wie sie sind ohne ihre Werte oder Struktur zu ändern.			
		DetectionProfile			
		<b>uuid</b> <u>UUID</u> (string), pattern: <code>^[a-f0-9-]+\$, <b>required</b>, read-only</code>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8		

Code	Body	JSON-Objekt			
		<b>alias</b> <i>Alias</i> (integer), <b>required</b> , read-only	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.		
		<b>name</b> <i>String</i> , <b>required</b>	Visuell lesbarer Name des Erkennungsprofils		
		<b>colorspace</b> <i>Colorspace</i> , <b>required</b>	Ein Farbraum beschreibt die numerische Umwandlung von Farben unter bestimmten Bedingungen. Verschiedene standardisierte Farbräume sind passend für verschiedene Erkennungsaufgaben.		
			<b>colorspace</b>		
			<b>name</b> <i>String</i> , <b>required</b>		
			<b>space_id</b> <i>ColorspaceID</i> , <b>required</b>	Eindeutiger Name eines Farbraums	
			<b>axes</b> Array of <i>ColorspaceAxis</i> , minimum items: 3, maximum items: 3, <b>required</b>	<i>ColorspaceAxis</i> []	
				<b>id</b> <i>String</i> , <b>required</b>	Eindeutiger Name
				<b>label</b> <i>String</i> , <b>required</b>	Visuell lesbarer Name
				<b>minimum</b> Number, <b>required</b>	Niedrigster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
				<b>maximum</b> Number, <b>required</b>	Höchster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
		<b>non_matching_output</b> <i>WantedSwitchingOutputsState</i> , <b>required</b>	Dieser Zustand der Schaltausgänge wird angewandt, wenn die aktuelle Stichprobenfarbe zu keiner gespeicherten Farbgruppe (Matcher) gehört.		
			<b>non_matching_output</b>		
			<b>uuid</b> <i>UUID</i> (string), pattern: ^[a-f0-9-]+\$, <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
			<b>states</b> Array of any of boolean or null, <b>required</b>	Liste von True/False/Null Werten be-	

Code	Body	JSON-Objekt			
					schreibt den gewollten Zustand des Schaltausgangs
		<b>non_matching_hold_time</b> HoldTime (number), maximum 3153600000, <b>required</b>	Minimale Dauer (in Sekunden) der Anwendung des <i>non_matching_output</i> Zustands am Schaltausgang des Sensors. Diese Verlängerung eines potentiellen <i>nicht passenden</i> Falls könnte nützlich sein, wenn die Bearbeitungszeit eines verbundenen Aktors den Stichprobenzeitraum des Sensors überschreitet.		
		<b>compensation_settings</b> CompensationSettings, <b>required</b>	Die Kompensationseinstellungen eines Erkennungsprofils beschreiben die Konfiguration eines internen Sensorkomponenten, bezogen auf die Stabilisations- und Kompensationsalgorithmen. Diese Werte können mittels Erteilung einer <b>POST</b> Anfrage entgegen <code>/api/sensor/detection-profiles/current/autogain</code> bestimmt werden. Das Ergebnis ist ein passendes Set aus Kompensationseinstellungen für diesen Sensor unter den aktuellen Umständen. Der Inhalt dieses Datenobjekts ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).		
			compensation_settings		
		<b>sampling_settings</b> SamplingSettings, <b>required</b>	Stichprobeneinstellungen beschreiben alle Details der Stichprobenentnahme. Ihre Attribute können abgefragt und untersucht werden (z.B. um die aktuelle Abtastrate abzurufen). Die meisten Werte die in den Stichprobeneinstellungen gespeichert sind sollten nicht direkt geändert werden. Der verbundene API Endpunkt <code>/api/sensor/detection-profiles/current/autogain</code> sollte stattdessen benutzt werden. Das einzige veränderbare Attribut innerhalb der Stichprobeneinstellungen ist der Durchschnittswert. Es ist sicher ihn zu ändern, obwohl die berechneten Standardwerte während eines Autogain-Betriebs optimal für die meisten Erkennungsaufgaben sein sollten.		
			sampling_settings		
			<b>led_intensity</b> Number, minimum: 0, maximum: 1, <b>required</b>		Relative Intensität des Emitters während der Leuchtphase

Code	Body	JSON-Objekt		
			<b>base_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Die Basisabtastrate bestimmt die Dauer der Abtastperiode. Nach jeder Abtastperiode, werden die erfassten Daten verarbeitet und ein neues Erfassungsergebnis wird berechnet (z.B. die passendste Farbgruppe (Matcher) für das gegebene Beispiel). Dies könnte den Zustand der Schaltausgänge beeinträchtigen oder konfigurierte Aktionen auslösen. Somit definiert die Basisabtastrate die Maximumrate der Änderungen für die Schaltausgänge. Siehe ebenso die effektive Abtastrate.
			<b>effective_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Die effektive Abtastrate ist ein numerisches Produkt der Basisabtastrate und der Anzahl der Durchschnitte. Sie bestimmt die Minimumdauer die ein Ziel benötigt um abgetastet zu werden um dessen visuellen Auftritt korrekt zu bestimmen. Mit dem Einstellen der Standardwerte auf eins, ist dieser Wert gleich zu der Basisabtastrate.
			<b>minimum_wanted_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Dieser informative Wert repräsentiert die Abtastrate, welche während des aktuellsten Autogain-Vorgangs angefragt wurde. Die effektive Abtastrate könnte von der gewollten Abtastrate abweichen, falls die angefragte Abtastrate aufgrund der Grenzen des Sensors nicht erreichbar war (z.B. Überschreiten der

Code	Body	JSON-Objekt			
				unterstützten Abtastrate) oder aufgrund der Umwelt (z.B. nicht genug Licht, sodass eine langsamere Amplifikation mit höherer Verstärkung nötig war).	
			<b>sample_light_phase</b> Boolean, <b>required</b>	Definiert ob der Sensor die internen Emitter zum Abtasten periodisch aktivierten soll	
			<b>sample_dark_phase</b> Boolean, <b>required</b>	Definiert ob der Sensor die internen Emitter zum Abtasten periodisch deaktivieren soll.	
			<b>averages</b> AverageSampleCount (integer), minimum: 1, <b>required</b>	Anzahl vorheriger Stichproben die für jedes Stichprobenergebnis gemittelt werden sollen. Ein rollierender Mittelungsalgorithmus wird an den Stichproben angewandt.	
			<b>amplification</b> AmplificationLevel (integer), <b>required</b>	Der Amplifikationslevel gibt die interne Konfiguration eines Verstärkers an. Der Wert ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).	
		<b>white_reference</b> Array of number, <b>required</b>	Das Weißreferenzattribut wird benutzt um einen benutzerdefinierten Farbausgleich anzugeben. Sein Inhalt unterliegt dem internen Gebrauch. Somit sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte (z.B. /api/sensor/detection-profiles/{itemId}/white-reference).		
		<b>normalization_constant</b> Array of number, <b>required</b>	Normierungskonstanten sind verbunden mit der Weißreferenz. Ihr Inhalt unterliegt dem internen Gebrauch. Somit sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte		

Code	Body	JSON-Objekt			
			(z.B. /api/sensor/detection-profiles/{itemId}/white-reference).		
	<b>errors</b> Array of Error, <b>required</b>	Error[]			
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode		
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte		
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung		
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item				

### 1.2.2.21 Erkennungsprofil auslesen

**GET** / sensor / detection-profiles {itemId}

Gibt ein einziges Erkennungsprofil aus.

Anfrage

<b>Path Variables</b>
<b>itemId</b> String, <b>required</b>

Antwort

Code	Body	JSON-Objekt			
200	Properties (object)				
	<b>data</b> <a href="#">DetectionProfile</a> , <b>required</b>	Ein Erkennungsprofil erkennt ein komplettes Set aus Sensoreinstellungen für eine gegebene Erkennungsaufgabe. Mehrere Profile können gespeichert werden, um zwischen verschiedenen Erkennungsaufgaben zu wechseln oder für die inkrementelle Entwicklung eines verfeinerten Profils. Manche Attribute eines Erkennungsprofils legen interne Details des Sensors offen, welche indirekt durch andere Mittel bestimmt werden sollten. Diese Attribute werden nur oberflächlich beschrieben, da sie so behandelt werden sollen wie sie sind ohne ihre Werte oder Struktur zu ändern.			
		DetectionProfile			
		<b>uuid</b> <a href="#">UUID</a> (string), pattern: <code>^[a-f0-9-]+\$, <b>required</b>, read-only</code>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8		
		<b>alias</b>	Ein numerischer Wert, welcher benutzt werden		

Code	Body	JSON-Objekt			
		<i>Alias</i> (integer), <b>required</b> , read-only	kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.		
		<b>name</b> String, <b>required</b>	Visuell lesbarer Name des Erkennungsprofils		
		<b>colorspace</b> Colorspace, <b>required</b>	Ein Farbraum beschreibt die numerische Umwandlung von Farben unter bestimmten Bedingungen. Verschiedene standardisierte Farbräume sind passend für verschiedene Erkennungsaufgaben.		
			colorspace		
			<b>name</b> String, <b>required</b>		
			<b>space_id</b> ColorspaceID, <b>required</b>	Eindeutiger Name eines Farbraums	
			<b>axes</b> Array of ColorspaceAxis, minimum items: 3, maximum items: 3, <b>required</b>	ColorspaceAxis[]	
				<b>id</b> String, <b>required</b>	Eindeutiger Name
				<b>label</b> String, <b>required</b>	Visuell lesbarer Name
				<b>minimum</b> Number, <b>required</b>	Niedrigster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
				<b>maximum</b> Number, <b>required</b>	Höchster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
		<b>non_matching_output</b> WantedSwitchingOutputsState, <b>required</b>	Dieser Zustand der Schaltausgänge wird angewandt, wenn die aktuelle Stichprobenfarbe zu keiner gespeicherten Farbgruppe (Matcher) gehört.		
			non_matching_output		
			<b>uuid</b> <u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667	

Code	Body	JSON-Objekt			
				und ISO / IEC 9834-8	
			<b>states</b> Array of any of boolean or null, <b>required</b>	Liste von True/False/Null Werten beschreibt den gewollten Zustand des Schaltausgangs	
		<b>non_matching_hold_time</b> HoldTime (number), maximum 3153600000, <b>required</b>	Minimale Dauer (in Sekunden) der Anwendung des <i>non_matching_output</i> Zustands am Schaltausgang des Sensors. Diese Verlängerung eines potentiellen <i>nicht passenden</i> Falls könnte nützlich sein, wenn die Bearbeitungszeit eines verbundenen Aktors den Stichprobenzeitraum des Sensors überschreitet.		
		<b>compensation_settings</b> CompensationSettings, <b>required</b>	Die Kompensationseinstellungen eines Erkennungsprofils beschreiben die Konfiguration eines internen Sensorkomponenten, bezogen auf die Stabilisations- und Kompensationsalgorithmen. Diese Werte können mittels Erteilung einer <b>POST</b> Anfrage entgegen <code>/api/sensor/detection-profiles/current/autogain</code> bestimmt werden. Das Ergebnis ist ein passendes Set aus Kompensationseinstellungen für diesen Sensor unter den aktuellen Umständen. Der Inhalt dieses Datenobjekts ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).		
			compensation_settings		
		<b>sampling_settings</b> SamplingSettings, <b>required</b>	Stichprobeneinstellungen beschreiben alle Details der Stichprobenentnahme. Ihre Attribute können abgefragt und untersucht werden (z.B. um die aktuelle Abtastrate abzurufen). Die meisten Werte die in den Stichprobeneinstellungen gespeichert sind sollten nicht direkt geändert werden. Der verbundene API Endpunkt <code>/api/sensor/detection-profiles/current/autogain</code> sollte stattdessen benutzt werden. Das einzige veränderbare Attribut innerhalb der Stichprobeneinstellungen ist der		

Code	Body	JSON-Objekt		
			Durchschnittswert. Es ist sicher ihn zu ändern, obwohl die berechneten Standardwerte während eines Auto-gain-Betriebs optimal für die meisten Erkennungsaufgaben sein sollten.	
			sampling_settings	
			<b>led_intensity</b> Number, minimum: 0, maximum: 1, <b>required</b>	Relative Intensität des Emitters während der Leuchtphase
			<b>base_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Die Basisabtastrate bestimmt die Dauer der Abtastperiode. Nach jeder Abtastperiode, werden die erfassten Daten verarbeitet und ein neues Erfassungsergebnis wird berechnet (z.B. die passendste Farbgruppe (Matcher) für das gegebene Beispiel). Dies könnte den Zustand der Schaltausgänge beeinträchtigen oder konfigurierte Aktionen auslösen. Somit definiert die Basisabtastrate die Maximumrate der Änderungen für die Schaltausgänge. Siehe ebenso die effektive Abtastrate.
			<b>effective_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Die effektive Abtastrate ist ein numerisches Produkt der Basisabtastrate und der Anzahl der Durchschnitte. Sie bestimmt die Minimumdauer die ein Ziel benötigt um abgetastet zu werden um dessen visuellen Auftritt korrekt zu bestimmen. Mit dem Einstellen der Standardwerte auf eins, ist dieser Wert gleich zu der Basisabtastrate.
			<b>minimum_wanted_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Dieser informative Wert repräsentiert die Abtastrate, welche während des aktuellsten

Code	Body	JSON-Objekt			
				Autogain-Vorgang angefragt wurde. Die effektive Abtastrate könnte von der gewollten Abtastrate abweichen, falls die angefragte Abtastrate aufgrund der Grenzen des Sensors nicht erreichbar war (z.B. Überschreiten der unterstützten Abtastrate) oder aufgrund der Umwelt (z.B. nicht genug Licht, sodass eine langsamere Amplifikation mit höherer Verstärkung nötig war).	
			<b>sample_light_phase</b> Boolean, <b>required</b>	Definiert ob der Sensor die internen Emitter zum Abtasten periodisch aktiviert soll	
			<b>sample_dark_phase</b> Boolean, <b>required</b>	Definiert ob der Sensor die internen Emitter zum Abtasten periodisch deaktivieren soll.	
			<b>averages</b> AverageSampleCount (integer), minimum: 1, <b>required</b>	Anzahl vorheriger Stichproben die für jedes Stichprobenergebnis gemittelt werden sollen. Ein rollierender Mittelungsalgorithmus wird an den Stichproben angewandt.	
			<b>amplification</b> AmplificationLevel (integer), <b>required</b>	Der Amplifikationslevel gibt die interne Konfiguration eines Verstärkers an. Der Wert ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).	
		<b>white_reference</b> Array of number, <b>required</b>	Das Weißreferenzattribut wird benutzt um einen benutzerdefinierten Farbausgleich anzugeben. Sein Inhalt unterliegt dem internen Gebrauch. Somit		

Code	Body	JSON-Objekt			
				sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte (z.B. /api/sensor/detection-profiles/{itemId}/white-reference).	
		<b>normalization_constant</b> Array of number, required		Normierungskonstanten sind verbunden mit der Weißreferenz. Ihr Inhalt unterliegt dem internen Gebrauch. Somit sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte (z.B. /api/sensor/detection-profiles/{itemId}/white-reference).	
	<b>errors</b> Array of Error, required	Error[]			
		<b>code</b> String, optional		Maschinenlesbarer spezifischer Fehlercode	
		<b>mapping</b> String, optional		Eine Referenz zum Parameter welcher den Fehler verursachte	
		<b>message</b> String, optional		Visuell lesbare Fehlerbeschreibung	
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item				

### 1.2.2.22 Beginne Autogain-Prozedur

**POST** / sensor / detection-profiles / {itemId} / autogain

Führe die Autogain-Prozedur aus, um passende Stichprobenmerkmale für die aktuelle Umwelt zu bestimmen. Die resultierende Stichprobeneinrichtung wird automatisch angewandt. Diese neuen Einstellungen sind in Kraft, sobald die Antwort versendet ist.

Die Autogain-Prozedur leitet eine dynamische Neukalibrierung des internen Emitters und aller Kompensationsprozesse ein. Es resultiert in schnellen Wechseln oder Blinken des internen Emitters (falls aktiviert). Der Betrieb ist normalerweise innerhalb von 15 Sekunden fertig. Die Antwort wird versendet, nachdem alle zugehörigen Aktivitäten vollständig sind. Spätere Anfragen geben Werte basierend auf den angepassten Stichprobeneinstellungen aus.

#### Anfrage

<b>Path Variables</b>	
<b>itemId</b> String, required	
<b>Body</b>	<b>JSON-Objekt</b>
Properties (AutogainSettings)	
<b>level</b> Number, default: 0.8, minimum: 0.01, maximum: 1, optional	Soll-Wert für die Autogain-Prozedur
<b>minimum_sample_rate</b> SampleRate (number), minimum: 0.02, optional	Gewünschte Abtastrate (der Standard ist die aktuelle Abtastrate)
<b>enable_internal_emitter</b> Boolean, default: true, optional	Steuert die Kraft der internen Lichtquelle
<b>enable_ambient_light_compensation</b> Boolean, default: true, optional	Steuert das Umgebungslichtkompensationsverfahren. Diese Einstellung ist nur dann relevant, wenn enable_internal_emitter true gesetzt ist. Die Umgebungslicht-

	<p>kompensation führt zu einem gepulsten Gebrauch des internen Lichtemitters. Stichproben werden gesammelt für wechselnde Leucht und Dunkelphasen. Dies erlaubt die Berechnung einer Farbstichprobe des Ziels ausschließlich aller optischen Schnittstellen externer Lichtquellen. Sie sollten Umgebungslichtkompensation nicht ausschalten, solange der optische Pfad perfekt isoliert ist. Andernfalls beeinträchtigt externes Licht zwangsläufig die Farbstichprobenerhebung.</p>
<p><b>averages</b> AverageSampleCount (integer), minimum: 1, optional</p>	<p>Anzahl zu mittelnder vorheriger Stichproben für jedes Stichprobenergebnis.</p>

**Beispiele**

```
{
  "level": 0.7,
  "minimum_sample_rate": 1500,
  "enable_internal_emitter": true,
  "enable_ambient_light_compensation": true
}
```

**Antwort**

Code	Body	JSON-Objekt	
200 400	Properties (object)		
	<p>data <b>sampling_settings</b> SamplingSettings, <b>required</b></p>	<p>Stichprobeneinstellungen beschreiben alle Details der Stichprobenentnahme. Ihre Attribute können abgefragt und untersucht werden (z.B. um die aktuelle Abtastrate abzurufen). Die meisten Werte die in den Stichprobeneinstellungen gespeichert sind sollten nicht direkt geändert werden. Der verbundene API Endpunkt <code>/api/sensor/detection-profiles/current/autogain</code> sollte stattdessen benutzt werden. Das einzige veränderbare Attribut innerhalb der Stichprobeneinstellungen ist der Durchschnittswert. Es ist sicher ihn zu ändern, obwohl die berechneten Standardwerte während eines Autogain-Betriebs optimal für die meisten Erkennungsaufgaben sein sollten.</p>	
		SamplingSettings	
		<p><b>led_intensity</b> Number, minimum: 0, maximum: 1, <b>required</b></p>	Relative Intensität des Emitters während der Leuchtphase
		<p><b>base_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b></p>	Die Basisabtastrate bestimmt die Dauer der Abtastperiode. Nach jeder Abtastperiode, werden die erfassten Daten verarbeitet und ein neues Erfassungsergebnis wird berechnet (z.B. die passendste Farbgruppe (Matcher) für das gegebene Beispiel). Dies könnte den Zustand der Schaltgänge beeinträchtigen oder konfigurierte Aktionen auslösen. Somit definiert die Basisabtastrate die Maximumrate der Änderungen für die Schaltgänge. Siehe ebenso die effektive Abtastrate.
		<p><b>effective_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b></p>	Die effektive Abtastrate ist ein numerisches

Code	Body	JSON-Objekt	
			Produkt der Basisabtastrate und der Anzahl der Durchschnitte. Sie bestimmt die Minimumdauer die ein Ziel benötigt um abgetastet zu werden um dessen visuellen Auftritt korrekt zu bestimmen. Mit dem Einstellen der Standardwerte auf eins, ist dieser Wert gleich zu der Basisabtastrate.
		<b>minimum_wanted_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Dieser informative Wert repräsentiert die Abtastrate, welche während des aktuellsten Autogain-Vorgangs angefragt wurde. Die effektive Abtastrate könnte von der gewollten Abtastrate abweichen, falls die angefragte Abtastrate aufgrund der Grenzen des Sensors nicht erreichbar war (z.B. Überschreiten der unterstützten Abtastrate) oder aufgrund der Umwelt (z.B. nicht genug Licht, sodass eine langsamere Amplifikation mit höherer Verstärkung nötig war).
		<b>sample_light_phase</b> Boolean, <b>required</b>	Definiert ob der Sensor die internen Emitter zum Abtasten periodisch aktivieren soll
		<b>sample_dark_phase</b> Boolean, <b>required</b>	Definiert ob der Sensor die internen Emitter zum Abtasten periodisch deaktivieren soll.
		<b>averages</b> AverageSampleCount (integer), minimum: 1, <b>required</b>	Anzahl vorheriger Stichproben die für jedes Stichprobenergebnis gemittelt werden sollen. Ein rollierender Mittelungsalgorithmus wird an den Stichproben angewandt.
		<b>amplification</b> AmplificationLevel (integer), <b>required</b>	Der Amplifikationslevel gibt die interne Konfiguration eines Verstärkers an. Der Wert ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).
	<b>compensation_settings</b> CompensationSettings, <b>required</b>	Die Kompensationseinstellungen eines Erkennungsprofils beschreiben die Konfiguration eines internen Sensorkomponenten, bezogen auf die Stabilisations- und Kompensationsalgorithmen.	

Code	Body	JSON-Objekt	
		Diese Werte können mittels Erteilung einer <b>POST</b> Anfrage entgegen <code>/api/sensor/detection-profiles/current/autogain</code> bestimmt werden. Das Ergebnis ist ein passendes Set aus Kompensationseinstellungen für diesen Sensor unter den aktuellen Umständen. Der Inhalt dieses Datenobjekts ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).	
		<code>compensation_settings</code>	
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlercodes ausgeben</b> LCOL.autogain LCOL.autogain.invalid_target_level LCOL.autogain.invalid_sample_rate LPLC.validation.boolean		

### 1.2.2.23 Anfrage benutzerdefinierte Weißreferenz

**GET** / sensor / detection-profiles / {itemId} / white-reference

Überprüfe das Vorhandensein einer benutzerdefinierten Weißreferenz. Eine erfolgreiche Antwort (HTTP Status 200) gibt an, dass eine benutzerdefinierte Weißreferenz in Nutzung ist. Die *nicht gefunden* Antwort (HTTP Status 404) gibt an, dass stattdessen die Werkseinstellung-Weißreferenz benutzt wird.

Der detaillierte Inhalt der Antwort ist irrelevant. Stattdessen wird das verbundene `normalization_constants` Feld des Erkennungsprofils, basierend auf der aktuellen Weißreferenz angepasst.

#### Anfrage

Path Variables
<b>itemId</b> String, <b>required</b>

#### Antwort

Code	Eine benutzerdefinierte Weißreferenz ist in Nutzung.		
200 404	Body	JSON-Objekt	
	Properties (object)		
	<b>data</b> Array of number, <b>required</b>	Das Weißreferenzattribut wird benutzt um einen benutzerdefinierten Farbausgleich anzugeben. Sein Inhalt unterliegt dem internen Gebrauch. Somit sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte (z.B. <code>/api/sensor/detection-profiles/{itemId}/white-reference</code> ).	
	<b>errors</b>	Error[]	

	Array of Error, required		
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung

### 1.2.2.24 Festlegen einer benutzerdefinierten Weißreferenz

**POST** /sensor/detection-profiles/{itemId}/white-reference

Verwende eine benutzerdefinierte Weißreferenz für die Farbhandhabung des Sensors. Die aktuelle abgetastete Farbe wird für die Berechnung der Weißreferenz genutzt. Sie sollten ein neutrales, weißes Ziel für diese Aktion auswählen.

Bitte beachten Sie, dass das Ändern der Weißreferenz nicht sofort in Kraft tritt. Daher sollten Sie drei Sekunden warten, bevor Sie neue Werte abtasten.

#### Anfrage

<b>Path Variables</b>
<b>itemId</b> String, required

#### Antwort

Code	Body	JSON-Objekt	
200 406	Properties (object)		
	<b>data</b> Array of number, required	Das Weißreferenzattribut wird benutzt um einen benutzerdefinierten Farbausgleich anzugeben. Sein Inhalt unterliegt dem internen Gebrauch. Somit sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte (z.B. /api/sensor/detection-profiles/{itemId}/white-reference).	
	<b>errors</b> Array of Error, required	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlermeldungen ausgeben</b> LCOL.white_reference.too_dark		

### 1.2.2.25 Trigger-Aktionen auslesen

**GET** / sensor / action-triggers

Der Sensor kann so programmiert werden, dass er auf spezifische externe oder interne Signale reagiert. Die verfügbaren Aktionen können entweder durch Auslöseingabezeilen oder durch API-Anfragen ausgelöst werden. Dies erlaubt benutzerdefiniertes Verhalten, z.B. Lehren von Farben durch einen externen Knopf.

Mehrere *Trigger Aktionen* können erstellt werden. Jeder *Trigger Aktion* weist eine oder mehrere Aktionen einem Signal zu (siehe `trigger_sources` in `/api/sensor/capabilities`). Mehrere *Trigger Aktionen* können auf eine Signal bezogen sein (siehe unten Auftragsausführung für Details).

Die einem *Trigger Aktionen* zugewiesenen Aktionen, werden separat ausgewertet. Somit ist es möglich dieselbe Aktion (sogar mit denselben Parametern) mehrmals festzulegen. Die Aktionsliste für einen *Trigger Aktionen* kann leer sein.

Die Aktionen innerhalb eines einzigen *Trigger Aktions*, werden erfolgreich der Reihe der Elemente in der Liste nach ausgeführt. Die Auftragsausführung unter mehreren *Trigger Aktions*elementen ist undefiniert.

Auslösefälle welche einen *Zustandswechsel* beschreiben (z.B. `trigger_0_edge_rising`), werden nur einmal im Moment des Zustandswechsels ausgesendet. Somit werden verknüpfte Aktionen nur einmal pro Zustandswechsel ausgeführt.

Auslösefälle welche einen *Zustand* beschreiben (z.B. `trigger_0_level_low`), werden solange der Zustand aktiv ist, kontinuierlich ausgesendet. Die Aktionen eines *Trigger Aktions* die mit solch einem Auslösefall verbunden sind, werden periodisch ausgeführt. Nach jeder Ausführung eines solchen *Trigger Aktions*, werden weitere Ausführungen für einen Zeitraum von einer Sekunde oder bis zum nächsten Zustandswechsels (was immer zuerst kommt) übersprungen. Die einzige Ausnahme für diese periodisch ausgeführten Aktionen ist die `enable_switching_output` Aktion. Wenn diese Aktion einem *Zustand* zugewiesen ist, dann wird sie überdacht wann immer die Haltezeit der aktuell erkannten Farbgruppe überschritten wird (d.h. für Farbgruppen mit Haltezeit null: in jeder Abtastperiode).

Siehe *Aktionen* für eine Liste unterstützter Aktionen.

Anfrage

Query Parameters	
<b>event</b> <u>UUID</u> (string), pattern: <code>^[a-f0-9-]+\$, required, read-only</code>	Filter Trigger Aktionen nach dem gegebenen Fallnamen (z.B. <code>trigger_0_edge_rising</code> )

Antwort

Code	Body	JSON-Objekt		
200	Properties (object)			
	<b>data</b> Object, <b>required</b>	data		
		<b>action-triggers</b> Array of <u>ActionTrigger</u> , <b>required</b>	ActionTrigger[]	
			<b>uuid</b> <u>UUID</u> (string), pattern: <code>^[a-f0-9-]+\$, required, read-only</code>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
			<b>event</b> TriggerEvent-Name (string), <b>required</b>	Jeder Fallname angeboten von <code>/api/sensor/capabilities</code> (Attribut <code>trigger_sources</code> ) ist erlaubt.
			<b>actions</b> Array of Action, <b>required</b>	Liste der auszuführenden Aktionen nach dem gegebenem Fall
				Action[]
			<b>name</b> String, <b>required</b>	Eindeutiger Name der Aktion
			<b>arguments</b> Object, <b>required</b>	arguments
		<b>errors</b>	Error[]	

		Array of Error, <b>required</b>		
			<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
			<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
			<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung

### 1.2.2.26 Entferne mehrere oder alle Trigger-Aktionen

**DELETE** / sensor / action-triggers

Entferne eine Auswahl an Trigger Aktionenn, entweder basierend auf einem gegebenem Filterkriterium (falls unterstützt für diese Auswahl) oder entferne alle Trigger Aktionen von der Sammlung.

Alle Lösch-Anfragen resultieren in einer leeren Erfolgsrückmeldung (204). Dies ist sogar gültig für eine ungefilterte **DELETE**-Anfrage entgegen einer leeren Sammlung oder für eine gefilterte **DELETE**-Anfrage entgegen einer Sammlung ohne Trigger Aktionen die dem Filter entsprechen.

Anfrage

<b>Query Parameters</b>	
<b>event</b> UUID (string), pattern: ^[a-f0-9-]+\$ <b>, required, read-only</b>	Lösche alle Trigger Aktionen des zugehörigen gegebenen Fallnamen (z.B. <i>trigger_0_edge_rising</i> )

Antwort

<b>Code</b>	
204	Die leere Antwort weist auf Erfolg hin

### 1.2.2.27 Erstelle Trigger-Aktionen

**POST** / sensor / action-triggers

Erstellt eine neue Trigger-Aktion.

Alle unterstützten Datenattribute im Body der Anfrage sind optional.

Anfrage

<b>Body</b>	<b>JSON-Objekt</b>	
Properties ( <a href="#">Action Trigger</a> )	<b>uuid</b> UUID (string), pattern: ^[a-f0-9-]+\$ <b>, required, read-only</b>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
	<b>event</b> TriggerEventName (string), <b>required</b>	Jeder Fallname angeboten von /api/sensor/capabilities (Attribut <i>trigger_sources</i> ) ist erlaubt.
	<b>actions</b> Array of Action, <b>required</b>	Liste der auszuführenden Aktionen nach dem gegebenem Fall
		Action[]
	<b>name</b> String, <b>required</b>	Eindeutiger Name der Aktion
	<b>arguments</b> Object, <b>required</b>	arguments

Beispiele

```
{
  "uuid": "3f26aff4-8650-42a0-b319-51776c443fbc",
  colorSENSOR CFO
}
```

```

"event": "trigger_0_edge_falling",
"actions": [
  {
    "name": "enable_switching_output",
    "arguments": {}
  }
]
}
    
```

Antwort

Code	Body	JSON-Objekt		
200 400	Properties (object)			
	<b>data</b> <i>ActionTrigger, required</i>	Ein Trigger Aktionen weist ein gegebenes Set aus Aktionen einem Event zu. Am Ende jeder Abtastperiode werden alle Fälle bewertet. Alle singgemäßen Aktionen werden danach ausgeführt.		
		ActionTrigger		
		<b>uuid</b> <i>UUID (string), pattern: ^[a-f0-9-]+\$, required, read-only</i>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
		<b>event</b> <i>TriggerEvent-Name (string), required</i>	Jeder Fallname angeboten von /api/sensor/capabilities (Attribut trigger_sources) ist erlaubt.	
		<b>actions</b> <i>Array of Action, required</i>	Liste der auszuführenden Aktionen nach dem gegebenem Fall	
			Action[]	
			<b>name</b> <i>String, required</i>	Eindeutiger Name der Aktion
			<b>arguments</b> <i>Object, required</i>	arguments
	<b>errors</b> <i>Array of Error, required</i>	Error[]		
		<b>code</b> <i>String, optional</i>	Maschinenlesbarer spezifischer Fehlercode	
		<b>mapping</b> <i>String, optional</i>	Eine Referenz zum Parameter welcher den Fehler verursachte	
		<b>message</b> <i>String, optional</i>	Visuell lesbare Fehlerbeschreibung	
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.validation.collection_size_exceeded			

1.2.2.28 Lösche Trigger-Aktionen

**DELETE** / sensor / action-triggers / {itemId}

Löscht eine einzige Trigger-Aktion.

Anfrage

colorSENSOR CFO

<b>Path Variables</b>
<b>itemId</b> String, required

Antwort

<b>Code</b>	
204	Die leere Antwort weist auf Erfolg hin
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item

1.2.2.29 Trigger-Aktionen ändern

PUT / sensor / action-triggers / {itemId}

Ändert einen einzigen Trigger Aktionen.

Anfrage

<b>Path Variables</b>		
<b>itemId</b> String, required		
<b>Body</b>	<b>JSON-Objekt</b>	
Properties (Action Trigger)		
<b>uuid</b> UUID (string), pattern: ^[a-f0-9-]+\$, required, read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
<b>event</b> TriggerEventName (string), required	Jeder Fallname angeboten von /api/sensor/capabilities (Attribut trigger_sources) ist erlaubt.	
<b>actions</b> Array of Action, required	Liste der auszuführenden Aktionen nach dem gegebenem Fall	
	Action[]	
	<b>name</b> String, required	Eindeutiger Name der Aktion
	<b>arguments</b> Object, required	arguments

Beispiele

```
{
  "uuid": "3f26aff4-8650-42a0-b319-51776c443fbc",
  "event": "trigger_0_edge_falling",
  "actions": [
    {
      "name": "enable_switching_output",
      "arguments": {}
    }
  ]
}
```

Antwort

<b>Code</b>	<b>Body</b>	<b>JSON-Objekt</b>		
200 400 404	<b>Properties (object)</b>			
	<b>data</b> ActionTrigger, required	Ein Trigger Aktionen weist ein gegebenes Set aus Aktionen einem Event zu. Am Ende jeder Abtastperiode werden alle Fälle		

Code	Body	JSON-Objekt		
		bewertet. Alle sinnngemäßen Aktionen werden danach ausgeführt.		
		ActionTrigger		
		<b>uuid</b> UUID (string), pattern: ^[a-f0-9-]+\$, <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
		<b>event</b> TriggerEvent- Name (string), <b>required</b>	Jeder Fallname angeboten von /api/sensor/capabilities (Attribut trigger_sources) ist erlaubt.	
		<b>actions</b> Array of Action, <b>required</b>	Liste der auszuführenden Aktionen nach dem gegebenem Fall	
			Action[]	
			<b>name</b> String, <b>required</b>	Eindeutiger Name der Aktion
			<b>arguments</b> Object, <b>required</b>	arguments
	<b>errors</b> Array of Error, <b>required</b>	Error[]		
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode	
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte	
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung	
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item			

### 1.2.2.30 Trigger-Aktionen auslesen

**GET** / sensor / action-triggers / {itemId}

Gibt eine einzige Trigger-Aktion aus.

Anfrage

Path Variables
<b>itemId</b> String, <b>required</b>

Antwort

Code	Body	JSON-Objekt		
200	<b>Properties (object)</b>			
	<b>data</b> ActionTrigger, <b>required</b>	Ein Trigger Aktionen weist ein gegebenes Set aus Aktionen einem Event zu. Am Ende jeder Abtastperiode werden alle Fälle bewertet. Alle sinnngemäßen Aktionen werden danach ausgeführt.		

Code	Body	JSON-Objekt		
		ActionTrigger		
		<b>uuid</b> UUID (string), pattern: <code>^[a-f0-9-]+\$, required,</code> read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
		<b>event</b> TriggerEvent-Name (string), <b>required</b>	Jeder Fallname angeboten von <code>/api/sensor/capabilities</code> (Attribut <code>trigger_sources</code> ) ist erlaubt.	
		<b>actions</b> Array of Action, <b>required</b>	Liste der auszuführenden Aktionen nach dem gegebenem Fall	
			Action[]	
			<b>name</b> String, <b>required</b>	Eindeutiger Name der Aktion
			<b>arguments</b> Object, <b>required</b>	arguments
	<b>errors</b> Array of Error, <b>required</b>	Error[]		
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode	
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte	
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung	
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item			

### 1.2.2.31 Controller Leistungsumfang auslesen

**GET** / sensor / capabilities

Antwort

Code	Body	JSON-Objekt		
200	<b>Properties (object)</b>			
	<b>data</b> SensorCapabilities, <b>required</b>	Stellt Zugang zu sensorischen Details, die von diesem Gerät unterstützt werden zur Verfügung (z.B. Farbraum, Eingabe- und Ausgabezeilen, ...).		
		SensorCapabilities		
		<b>maximum_sample_rate</b> Integer, <b>required</b>	Die maximale Abtastrate die der Sensor unterstützt	
		<b>tolerances</b> Array of ColorTolerance (union), <b>required</b>	Liste aus Toleranzspezifikationen welche vom Sensor unterstützt werden	
			InfiniteColorTolerance	
			<b>limits</b> Object, <b>required</b>	limits
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities .</code>
			SphereColorTolerance	
			<b>limits</b>	limits

Code	Body	JSON-Objekt			
			Object, <b>required</b>	radius Numer, <b>required</b>	
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via /api/sensor/capabilities .	
			CylinderColorTolerance		
			<b>limits</b> Object, <b>required</b>	limits <b>radius</b> Number, <b>required</b> <b>half_height</b> Number, <b>required</b>	
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via /api/sensor/capabilities .	
			BoxColorTolerance		
			<b>limits</b> Object, <b>required</b>	limits <b>half_edges</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	
			<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via /api/sensor/capabilities .	
		<b>output_drivers</b> Array of SwitchingOutputDriver (string), <b>required</b>	Liste der unterstützten elektrischen Ausgangstreiber.		
		<b>trigger_sources</b> Array of TriggerSource, <b>required</b>	Beinhaltet die Liste verfügbarer Auslösequellen mit ihrem dazugehörigen Auslösefall. Auslösefälle können zum Ausführen bestimmter Aktionen automatisiert werden.		
			TriggerSource[]		
			<b>name</b> String, <b>required</b>	Name des Auslöseeingangs	
			<b>events</b> Array of TriggerEvent, <b>required</b>	TriggerEvent []	
				<b>name</b> TriggerEventName (string), <b>required</b>	
		<b>output_pin_count</b> Integer, <b>required</b>	Anzahl verfügbarer Schaltausgangszeilen		
		<b>actions</b> Array of Action, <b>required</b> , Deprecated	Veraltet: benutze stattdessen /api/actions		
			Action[]		
			<b>name</b> String, <b>required</b>	Eindeutiger Name einer Aktion	
			<b>arguments</b> Object, <b>required</b>	arguments	
		<b>colorspaces</b> Array of Colorspace, <b>required</b>	Liste unterstützter Farbräume.		
			Colorspace[]		
			<b>name</b> String, <b>required</b>		

Code	Body	JSON-Objekt			
			<b>space_id</b> ColorspaceID, <b>required</b>	Eindeutiger Name eines Farbraums	
			<b>axes</b> Array of ColorspaceAxis, minimum items: 3, maximum items: 3, <b>required</b>	ColorspaceAxis[]	
				<b>id</b> String, <b>required</b>	Eindeutiger Name
				<b>label</b> String, <b>required</b>	Visuell lesbarer Name
				<b>minimum</b> Number, <b>required</b>	Niedrigster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
				<b>maximum</b> Number, <b>required</b>	Höchster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
		<b>colorspace_tolerance_maps</b> Array of ColorspaceToleranceMap, <b>required</b>	Die Auswertung der Toleranzen entgegen der Positionen der Farben, hängt von dem aktuell eingestellten Farbraum ab. Zum Beispiel das Toleranzattribut „half_height“ bezieht sich auf die Helligkeitsbezogene Achse eines Farbraums (z.B. „L*“ für den „Lab*“ Farbraum) und wird benutzt für die Höhe der zylindrischen Toleranzform und der ersten Ecke der Box-Toleranzform. Die Farbton-bezogenen Attribute (z.B. „a“ und „b“ für den „Lab*“ Farbraum) werden für den „Radius“ einer Zylinder-Toleranzform und für die zweite und dritte Ecke der Box-Toleranzform benutzt. Die <i>colorspace_tolerance_maps</i> legen diese Beziehungen zwischen Farbräumen und Toleranzen fest.		
			ColorspaceToleranceMap[]		
			<b>colorspace_id</b> ColorspaceID (string), <b>required</b>	Eindeutiger Name eines Farbraums	
			<b>tolerance_shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .	
			<b>limits_axes_map</b>	limits_axes_map	

Code	Body	JSON-Objekt			
			Object, <b>required</b>		
				<b>half_height</b> Array of string, optional	
				<b>half_edges</b> Array of string, optional	
				<b>radius</b> Array of string, optional	
		<b>settings_categories</b> Array of string, <b>required</b>	Liste der Kategorien, welche während eines Imports ausgewählt werden können, um zu steuern welche Einstellungen angewandt werden sollen. Siehe die Dokumentation für die <b>POST</b> -Anfrage auf <code>/api/settings</code>		
		<b>maximum_detectables_count</b> Integer, <b>required</b>	Maximale Anzahl an Farbpositionen (Detectable), welche in einem Erkennungsprofil gespeichert werden können.		
		<b>maximum_matchers_count</b> Integer, <b>required</b>	Maximale Anzahl an Erkennungsergebnissen (Matcher), welche in einem Erkennungsprofil gespeichert werden können.		
	<b>errors</b> Array of Error, <b>required</b>	Error[]			
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode		
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte		
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung		

### 1.2.2.32 Farbräume auslesen

**GET** / sensor / colorspaces

Gibt eine Liste verfügbarer Farbräume aus.

Antwort

Code	Body	JSON-Objekt			
200	<b>Properties (object)</b>				
	<b>data</b> Object, <b>required</b>	data			
		<b>colorspaces</b> Array of Colorspace, <b>required</b>	Colorspace[]		
			<b>name</b> String, <b>required</b>		
			<b>space_id</b> ColorspaceID, <b>required</b>	Eindeutiger Name eines Farbraums	
			<b>axes</b> Array of ColorspaceAxis, minimum items: 3, maximum items: 3, <b>required</b>	ColorspaceAxis[]	

Code	Body	JSON-Objekt			
				<b>id</b> String, <b>required</b>	Eindeutiger Name
				<b>label</b> String, <b>required</b>	Visuell lesbarer Name
				<b>minimum</b> Number, <b>required</b>	Niedrigster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
				<b>maximum</b> Number, <b>required</b>	Höchster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
	<b>errors</b> Array of Error, <b>required</b>	Error[]			
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode		
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte		
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung		

### 1.2.2.33 Farbraum abrufen

**GET** / sensor / colorspaces / {space\_id}

Gibt einen einzigen Farbraum aus.

Anfrage

Path Variables
<b>space_id</b> String, <b>required</b>

Antwort

Code	Body	JSON-Objekt		
200	Properties (object)			
	<b>data</b> Colorspace, <b>required</b>	Ein Farbraum beschreibt die numerische Umwandlung von Farben unter bestimmten Bedingungen. Verschiedene standardisierte Farbräume sind passend für verschiedene Erkennungsaufgaben.		
		colorspace		
		<b>name</b> String, <b>required</b>		
		<b>space_id</b> ColorspaceID, <b>required</b>	Eindeutiger Name eines Farbraums	
		<b>axes</b> Array of Colorspace-Axis, minimum items: 3, maximum items: 3, <b>required</b>	ColorspaceAxis[]	
			<b>id</b> String, <b>required</b>	Eindeutiger Name
			<b>label</b> String, <b>required</b>	Visuell lesbarer Name

Code	Body	JSON-Objekt		
			<b>minimum</b> Number, <b>required</b>	Niedrigster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
			<b>maximum</b> Number, <b>required</b>	Höchster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
	<b>errors</b> Array of Error, <b>required</b>	Error[]		
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode	
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte	
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung	
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item			

### 1.2.3 Einstellungen

Verwaltung aller Geräteeinstellungen

#### 1.2.3.1 Einstellungen exportieren

**GET** / settings

Exportiere alle Konfigurationen des Geräts.

Antwort

Code	Body	
200	<p>Textliche Darstellung der kompletten Gerätekonfiguration. Dieser Konfigurationsexport kann ohne Änderungen auf denselben oder einen anderen Sensor hochgeladen werden.</p> <p>Die Konfigurationsdaten sind als Base64 kodiert. Die Base64-Kodierung soll signalisieren, dass der Konfigurationsdaten-Dump nicht dazu gedacht ist, automatisch manipuliert oder kontrolliert zu werden. Sie dürfen sich nicht auf eine bestimmte interne Struktur verlassen, da sie sich im Laufe der Zeit ohne weitere Ankündigung ändern kann.</p>	
	<b>Body</b>	<b>text/plain</b>

#### 1.2.3.2 Einstellungen hochladen

**POST** / settings

Ersetze die Gerätekonfiguration mit denen, die hochgeladen werden.

Wenn Sie die Einstellungen nur teilweise importieren wollen, können Sie das machen indem Sie eine oder mehrere Import-Kategorien angeben. Wenn Sie nicht mindestens eine Import-Kategorie angeben, dann ist der Standard, dass alle importiert werden.

Import-Kategorie Schlüssel	Importiert
import_category_access	Benutzer, Rollen und Berechtigungen
import_category_firmware	Firmware-Einstellungen (wie den Zweig, aber nicht die Firmware an sich!)
import_category_keybad	Tastatureinstellungen
import_category_network	Netzwerkconfiguration
import_category_outputs	Treiber Ausgang

import_category_sensor	Farbgruppen (Matchers), Farben (Detectables), Farbraum, Abtastkonfiguration
import_category_system	Systemeinstellungen (wie Zeitzone und Hostname)

Eine maschinenlesbare Liste an Import-Kategorien wird ausgegeben, durch den `/api/sensor/capabilities` Endpunkt unter dem `settings_categories` Schlüssel.

Einstellungen welche aus älteren Firmwares exportiert wurden, werden automatisch auf das neue Format, welches der Sensor benötigt übertragen. Falls eine Übertragung scheitern sollte, enthält die Antwort den Fehlercode `LPLC.migration.execution_failed`. Einstellungen aus aktuelleren Firmwares als diejenige welche am Sensor benutzt wird, könnten im Import scheitern, wenn das Einstellungsformat nicht weiterhin kompatibel ist. In diesem Fall enthält die Antwort den Fehlercode `LPLC.migration.future_version`.

Anfrage

<b>Body</b>	<b>multipart/form-data</b>
Properties (object)	
<b>settings_file</b> File, required	Die Einstellungen welche einen Base64-verschlüsselten Konfigurations-Dump enthalten. Siehe GET-Anfrage für <code>/api/settings</code> .
<code>/^import_category_+/ Any, optional</code>	Importiere nur eine spezifische Teilmenge der Konfiguration.

Antwort

<b>Code</b>	
204	Die leere Antwort weist auf Erfolg hin.
400	<b>Könnte folgende Fehlercodes enthalten</b> <code>LPLC.format.encoding.utf8</code> <code>LPLC.format.malformed.base64</code> <code>LPLC.validation.missing_input</code> <code>LPLC.format.malformed.json</code> <code>LPLC.format.malformed.json.not_dict</code> <code>LPLC.migration.future_version</code> <code>LPLC.migration.execution_failed</code>

1.2.3.3 Einstellungen importieren

**PUT** / settings

Ersetze die komplette Gerätekonfiguration mit dem hochgeladenen Konfigurations-Dump.

Der Inhalt der hochgeladen wird, kann über eine `GET`-Anfrage auf `/api/settings` abgerufen werden. Dieser Base64-verschlüsselte Konfigurations-Dump wird im Anfrage-Body angenommen.

Einstellungen welche aus älteren Firmwares exportiert wurden, werden automatisch auf das neue Format, welches der Sensor benötigt übertragen. Falls eine Übertragung scheitern sollte, enthält die Antwort den Fehlercode `LPLC.migration.execution_failed`. Einstellungen aus aktuelleren Firmwares als diejenige welche am Sensor benutzt wird, könnten im Import scheitern, wenn das Einstellungsformat nicht weiterhin kompatibel ist. In diesem Fall enthält die Antwort den Fehlercode `LPLC.migration.future_version`.

Anfrage

<b>Body</b>	<b>text/plain</b>
-------------	-------------------

Antwort

<b>Code</b>	
-------------	--

204	Die leere Antwort weist auf Erfolg hin.
400	<b>Könnte folgende Fehlercodes enthalten</b> LPLC.format.encoding.utf8 LPLC.format.malformed.base64 LPLC.format.malformed.json LPLC.format.malformed.json.not_dict LPLC.migration.future_version LPLC.migration.execution_failed

### 1.2.3.4 Einstellungen zurücksetzen

**DELETE** / settings

Setze die Gerätekonfiguration auf Werkeinstellungen zurück.

Antwort

Code	
204	Die leere Antwort weist auf Erfolg hin.

### 1.2.4 System

Verwaltung der Systemeinstellungen des Geräts.

#### 1.2.4.1 Systemeinstellungen anfragen

**GET** / system

Antwort

Code	Body	JSON-Objekt	
200	Properties (object)		
	<b>data</b> SystemSettings, required	SystemSettings	
		<b>hostname</b> Hostname, pattern: ^(?:[a-zA-Z0-9](?:[a-zA-Z0-9\-\ ]*[a-zA-Z0-9])?\.)*[a-zA-Z0-9](?:[a-zA-Z0-9\-\ ]*[a-zA-Z0-9])?\$/, optional	Visuell lesbarer Name, welcher den das Gerät im Netzwerk identifiziert.
		<b>uptime</b> any of number or null, optional, read-only	Die aktuelle Systemverfügbarkeit in Sekunden. Obwohl höchst unwahrscheinlich, kann es null sein, falls das System einen ungültigen Wert meldet.
	<b>errors</b> Array of Error, required	Error []	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung

#### 1.2.4.2 Systemeinstellungen ändern

**PUT** / system

Anfrage

Body	JSON-Objekt
Properties (SystemSettings)	
<b>hostname</b>	Visuell lesbarer Name, welcher den das Gerät im Netzwerk identifiziert.

Hostname, pattern: ^(?:[a-zA-Z0-9](?:[a-zA-Z0-9\-\ ]*[a-zA-Z0-9])?\.)*[a-zA-Z0-9](?:[a-zA-Z0-9\-\ ]*[a-zA-Z0-9])?\$/, optional	
<b>uptime</b> any of number or null, optional, read-only	Die aktuelle Systemverfügbarkeit in Sekunden. Auch wenn es höchst unwahrscheinlich ist, kann der Wert mal null sein, falls das System einen ungültigen Wert meldet.

**Beispiele**

```
{
  "hostname": "cfo-7454232361"
}
```

**Antwort**

Code	Body	JSON-Objekt	
200 400 500	<b>Properties (object)</b>		
	<b>data</b> SystemSettings, required	SystemSettings	
		<b>hostname</b> Hostname, pattern: ^(?:[a-zA-Z0-9](?:[a-zA-Z0-9\-\ ]*[a-zA-Z0-9])?\.)*[a-zA-Z0-9](?:[a-zA-Z0-9\-\ ]*[a-zA-Z0-9])?\$/, optional	Visuell lesbarer Name, welcher den das Gerät im Netzwerk identifiziert.
		<b>uptime</b> any of number or null, optional, read-only	Die aktuelle Systemverfügbarkeit in Sekunden. Kann höchst unwahrscheinlich null sein kann, falls das System einen ungültigen Wert meldet.
	<b>errors</b> Array of Error, required	Error []	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.system.action_failed		

**1.2.4.3 Zurücksetzen auf Werksfirmware oder -einstellungen**

**POST** / system / factory-reset

Setze die Firmware des Sensors auf Werkseinstellungen zurück und leite einen Neustart ein. Nach der Fertigstellung, nutzt der Sensor seine originale („Wiederherstellungs-“) Firmware und alle Einstellungen werden auf Standard zurückgesetzt. Die Firmware-Wiederherstellung kann durch "/firmware/upgrade-from-current" aktualisiert werden.

Wenn Sie nur die Einstellungen zurücksetzen wollen, ist es ausreichend eine **DELETE**-Anfrage an den /api/settings Endpunkt zu senden.

**Antwort**

<b>Code</b>	
204 500	Die leere Antwort weist auf Erfolg hin.
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.system.action_failed

### 1.2.4.4 Neustart durchführen

**POST** / system / reboot

Startet das Gerät neu.

Der Software-ausgelöste Neustart ist, im Vergleich zum Herausziehen der Stromversorgung die höflichere Methode um den Sensor abzuschalten. Allerdings ist Letzteres genauso sicher.

Antwort

Code	
204 500	Die leere Antwort weist auf Erfolg hin.
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.system.action_failed

### 1.2.4.5 Zeiteinstellungen auslesen

**GET** / system / time

Antwort

Code	Body	JSON-Objekt	
200	Properties (object)		
	<b>data</b> SystemTimeSettings, <b>required</b>	SystemTimeSettings	
		<b>now</b> Timestamp (string), optional	Aktuelle Zeit aus der Sicht des Sensors
		<b>timezone</b> String, optional	Aktuell konfigurierte Zeitzone
		<b>ntp_servers</b> Array of string, optional	Ein oder mehrere Netzwerk-Zeitserver
		<b>default_ntp_servers</b> Array of string, optional, read-only	Vorkonfigurierte Netzwerk-Zeitserver
	<b>errors</b> Array of Error, <b>required</b>	Error []	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung

### 1.2.4.6 Zeiteinstellungen ändern

**PUT** / system / time

Anfrage

Body	JSON-Objekt
Properties (SystemTimeSettings)	
<b>now</b> Timestamp (string), optional	Aktuelle Zeit aus der Sicht des Sensors
<b>timezone</b> String, optional	Aktuell konfigurierte Zeitzone
<b>ntp_servers</b> Array of string, optional	Ein oder mehrere Netzwerk-Zeitserver
<b>default_ntp_servers</b> Array of string, optional, read-only	Vorkonfigurierte Netzwerk-Zeitserver

**Beispiele**

```
{
  "now": "2018-01-24T15:45:15.694004+01:00",
  "timezone": "Europe/Berlin",
  "ntp_servers": [
    "pool.ntp.org"
  ],
  "default_ntp_servers": [
    "pool.ntp.org"
  ]
}
```

**Antwort**

Code	Body	JSON-Objekt	
200 400 500	Properties (object)		
	<b>data</b> SystemTimeSettings, <b>required</b>	SystemTimeSettings	
		<b>now</b> Timestamp (string), optional	Aktuelle Zeit aus der Sicht des Sensors
		<b>timezone</b> String, optional	Aktuell konfigurierte Zeitzone
		<b>ntp_servers</b> Array of string, optional	Ein oder mehrere Netzwerk-Zeitserver
		<b>default_ntp_servers</b> Array of string, optional, read-only	Vorkonfigurierte Netzwerk-Zeitserver
	<b>errors</b> Array of Error, <b>required</b>	Error []	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.system.action_failed LPLC.validation.readonly LPLC.validation.readonly		

**1.2.4.7 Unterstützte Zeitzonen abrufen**

**GET** / system / time / zones

Das Gerät beinhaltet Wissen über eine vollständige Liste aus offiziellen standardisierten Zeitzonen. Der Sensor sollte entweder mit einer lokalen Zeitzone oder mit UTC konfiguriert werden.

**Antwort**

Code	Body	JSON-Objekt	
200	Properties (object)		
	<b>data</b> Object, <b>required</b>	Data	
		<b>Timezone_names</b> Array of string, <b>required</b>	Liste aus vom Gerät unterstützten Zeitzonen
	<b>errors</b> Array of Error, <b>required</b>	Error []	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte

		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
--	--	------------------------------------	------------------------------------

### 1.2.5 Netzwerk

Die Services des Sensorgeräts sind durch Netzwerkverbindungen zugänglich. Die Netzwerkschnittstellen des Geräts können für alle normgerechten Netzwerk-Setups konfiguriert werden.

#### 1.2.5.1 Netzwerkeinstellungen zurücksetzen

**DELETE** / network

Setze Netzwerkeinstellungen auf Werkseinstellungen zurück.

Antwort

Code	
204	Die leere Antwort weist auf Erfolg hin

#### 1.2.5.2 Status und Konfiguration der Netzwerkschnittstellen auslesen

**GET** / network / interfaces

Gibt eine Liste Netzwerkschnittstellen aus.

Jede Netzwerkschnittstelle hat eine MAC-Adresse (`hardware_address`), einen eindeutigen Namen (`iface`) und gibt den aktuellen, physischen Verbindungsstatus (`has_link`) an.

Beide IPv4 (`ipv4`) und IPv6 (`ipv6`) werden unterstützt, beide mit ihrer aktuellen Konfiguration (`address_configuration`) und aktueller Schnittstellenadressen (`current_addresses`).

Schnittstellenadressen sind geschrieben in CIDR-Notation, startend mit der Schnittstellenadresse, gefolgt von einem Schrägstrich und endend mit einer Dezimalzahl, welche die Subnetzmaske (IPv4) oder Präfix-Länge (IPv6).

Die Sammlung an sich ist schreibgeschützt, damit keine neuen Schnittstellen hinzugefügt werden können. Sie können die individuelle Schnittstellenkonfiguration mit einer PUT-Anfrage zu einer spezifischen Schnittstellenquelle ändern.

Antwort

Code	Body	JSON-Objekt			
200	Properties (object)				
	<b>data</b> Array of <code>NetworkInterfaceAddressConfigurationState</code> , <b>required</b>	<code>NetworkInterfaceAddressConfigurationState[]</code>			
		<b>ipv4</b> <code>NetworkInterfaceAddressFamilyStateIPv4</code> , optional	IPv4 Netzwerkadress-konfiguration		
			ipv4		
			<b>address_configurations</b> Array of any of <code>NetworkAddressConfigurationIPv4Static</code> or <code>NetworkAddressConfigurationIPv4DHCP</code> , optional	<code>NetworkAddressConfigurationIPv4Static[]</code>	
			<b>method</b> string, one of <code>[static, dhcp]</code> , <b>required</b>		Konfigurationsmethode, welche für die Adresse benutzt wird.

Code	Body	JSON-Objekt			
				<b>address</b> <a href="#">NetworkInterfaceAddressIPv4</a> (string), <b>required</b>	IPv6 Netzwer- kadresse in CIDR-Notation
				<b>gateway</b> <a href="#">NetworkAddressIPv4</a> (string), optional	Standard Ga- teway für ab- gehenden Verkehr
				NetworkAddressConfi- gurationIPv4DHCP[]	
				<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurati- onsmethode, welche für die Adresse be- nutzt wird.
		<b>ipv6</b> <a href="#">NetworkInterfaceAddress- FamilyStateIPv6</a> , optional	IPv6 Netzwerkadress- konfiguration		
			ipv6		
			<b>address_configurations</b> Array of any of <a href="#">Net- workAddressConfigura- tionIPv6Static</a> , <a href="#">Net- workAddressConfigura- tionIPv6DHCP</a> or <a href="#">Net- workAddressConfigura- tionIPv6Auto</a> , optional	NetworkAddressConfi- gurationIPv6Static[]	
				<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurati- onsmethode, welche für die Adresse be- nutzt wird.
				<b>address</b> <a href="#">NetworkInter- faceAddressIPv4</a> (string), <b>required</b>	IPv6 Netzwer- kadresse in CIDR-Notation
				<b>gateway</b> <a href="#">NetworkAddressIPv4</a> (string), optional	Standard Ga- teway für ab- gehenden Verkehr
				NetworkAddressConfi- gurationIPv6DHCP[]	
				<b>method</b> string, one of [static, dhcp, auto], <b>re- quired</b>	Konfigurati- onsmethode, welche für die Adresse be- nutzt wird.
				NetworkAddressConfi- gurationIPv6Auto[]	
				<b>method</b> string, one of [static, dhcp, auto], <b>re- quired</b>	Konfigurati- onsmethode, welche für die Adresse be- nutzt wird.
	<b>errors</b> Array of Error, <b>required</b>	Error[]			
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehler- code		
		<b>mapping</b> String, optional	Eine Referenz zum Pa- rameter welcher den Fehler verursachte		
		<b>message</b> String, optional	Visuell lesbare Fehler- beschreibung		

### 1.2.5.3 Status und Konfiguration einer einzelnen Netzwerkschnittstelle auslesen

**GET** / network / interfaces / {name}

Gibt Informationen, aktuellen Status, Adresskonfiguration und aktuelle Adresse für eine einzige Schnittstelle aus.

Anfrage

<b>Path Variables</b>
<b>name</b> String, <b>required</b>

Antwort

Code	Body	JSON-Objekt			
200	Properties (object)				
	<b>data</b> Array of <a href="#">NetworkInterfaceAddressConfigurationState</a> , <b>required</b>	NetworkInterfaceAddressConfigurationState[]			
		<b>ipv4</b> <a href="#">NetworkInterfaceAddressFamilyStateIPv4</a> , optional	IPv4 Netzwerkadresskonfiguration		
			ipv4		
			<b>address_configurations</b> Array of any of <a href="#">NetworkAddressConfigurationIPv4Static</a> or <a href="#">NetworkAddressConfigurationIPv4DHCP</a> , optional	NetworkAddressConfigurationIPv4Static[]	
				<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
				<b>address</b> <a href="#">NetworkInterfaceAddressIPv4</a> (string), <b>required</b>	IPv6 Netzwerkadresse in CIDR-Notation
				<b>gateway</b> <a href="#">NetworkAddressIPv4</a> (string), optional	Standard Gateway für abgehenden Verkehr
				<a href="#">NetworkAddressConfigurationIPv4DHCP</a> []	
				<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
		<b>ipv6</b> <a href="#">NetworkInterfaceAddressFamilyStateIPv6</a> , optional	IPv6 Netzwerkadresskonfiguration		
			ipv6		
			<b>address_configurations</b> Array of any of <a href="#">NetworkAddressConfigurationIPv6Static</a> , <a href="#">NetworkAddressConfigurationIPv6DHCP</a> or <a href="#">NetworkAddressConfigurationIPv6Auto</a> , optional	NetworkAddressConfigurationIPv6Static[]	
				<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die

Code	Body	JSON-Objekt			
					Adresse benutzt wird.
				<b>address</b> <a href="#">NetworkInterfaceAddressIPv4</a> (string), <b>required</b>	IPv6 Netzwerkadresse in CIDR-Notation
				<b>gateway</b> <a href="#">NetworkAddressIPv4</a> (string), optional	Standard Gateway für abgehenden Verkehr
				NetworkAddressConfigurationIPv6DHCP[]	
				<b>method</b> string, one of [static, dhcp, auto], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
				NetworkAddressConfigurationIPv6Auto[]	
				<b>method</b> string, one of [static, dhcp, auto], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
	<b>errors</b> Array of Error, <b>required</b>	Error[]			
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode		
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte		
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung		
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item				

### 1.2.5.4 IPv4 und/oder IPv6 Adresskonfiguration

**PUT** / network / interfaces / {name}

Schnittstellen sind schreibgeschützt, außer ihre IPv6 und IPv4 Adresskonfiguration.

Sie können eine neue Adresskonfiguration festlegen durch Bereitstellen des `address_configurations` Schlüssels in dem jeweiligen IP-Adressraum (`ipv4` oder `ipv6`). `address_configurations` ersetzt jede existierende Konfiguration durch die neue Konfiguration. Wenn Sie nur eine neue Konfiguration hinzufügen wollen, seien Sie sich sicher bereits existierende ebenfalls einzureichen.

Die `address_configurations` Liste beeinflusst nur ihre jeweilige Adressenfamilie. Wenn Sie nur die IPv4 Adresskonfiguration ändern möchten, ist es ausreichend den `ipv4.address_configurations` Schlüssel festzulegen. Jede andere Adressenfamilie bleibt unberührt, solange Sie nicht ebenso ihre eigene `address_configuration` Liste ändern.

Die Antwort an eine Netzwerkkonfigurations-Änderungsanfrage wird ausgegeben *bevor* die neue Konfiguration angewandt wird. Dies stellt sicher, dass der Anrufer eine Anerkennung von der API bekommt, bevor die Netzwerkverbindung möglicherweise durch die veränderte Konfiguration unterbrochen wird. Die Neukonfiguration des neuen Netzwerk-Setups findet im Hintergrund, kurz nach der Antwortenübermittlung, statt. Somit könnte die API existierende Verbindungen schließen und auf weitere Anfragen für einige Sekunden nicht mehr antworten. Bitte beachten Sie, dass nur eine Konfigurationsänderung gleichzeitig angefordert werden kann. Somit zögert die API die Antwort auf eine zweite Anfrage hinaus, bis alle internen Prozesse für die erste Anfrage beendet sind. So eine Antwort auf eine schnelle, nachfolgende Anfrage könnte auf bis zu 20 Sekunden hinausgezögert werden. Wenn zu viele parallele Anfragen

um eine Netzwerk-Setup-Konfiguration konkurrieren, wird die HTTP Statusantwort 423 (*gesperrt*) nach einer Zeitüberschreitung von 20 Sekunden zurückgegeben.

**Bitte beachten Sie:** Obwohl die API mehrere Adresskonfigurationen für jede Adressenfamilie unterstützt, wird nur die erste im Augenblick angewandt. Dies ist eine bevorstehende Funktion. Dafür sollten Sie nur ein Adresskonfigurationsitem pro Adressenfamilie zur Verfügung stellen.

Anfrage

Path Variables
<b>name</b> String, <b>required</b>

Body	JSON-Objekt		
Properties (NetworkInterfaceInformation)			
<b>iface</b> <a href="#">NetworkInterfaceName</a> (string), pattern: <code>^[a-z0-9]+\$</code> , <b>required</b> , read-only	Eindeutiger Name welcher Netzwerkschnittstelle beschreibt		
<b>hardware_address</b> <a href="#">MacAddress</a> (string), pattern: <code>^([a-f0-9]{2}:){5}[a-f0-9]{2}\$</code> , <b>required</b> , read-only	Eindeutige Hardwareadresse einer Netzwerkschnittstelle		
<b>has_link</b> boolean, <b>required</b> , read-only	Aktueller physikalischer Verbindungsstatus (ob ein Kabel eingesteckt ist oder nicht)		
<b>ipv4</b> <a href="#">NetworkInterfaceAddressFamilyStateIPv4</a> , optional	IPv4 Netzwerkadresskonfiguration		
	ipv4		
	<b>address_configurations</b> Array of any of <a href="#">NetworkAddressConfigurationIPv4Static</a> or <a href="#">NetworkAddressConfigurationIPv4DHCP</a> , optional	NetworkAddressConfigurationIPv4Static[]	
		<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
		<b>address</b> <a href="#">NetworkInterfaceAddressIPv4</a> (string), <b>required</b>	IPv6 Netzwerkadresse in CIDR-Notation
		<b>gateway</b> <a href="#">NetworkAddressIPv4</a> (string), optional	Standard Gateway für abgehenden Verkehr
		NetworkAddressConfigurationIPv4DHCP[]	
		<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
<b>ipv6</b> <a href="#">NetworkInterfaceAddressFamilyStateIPv6</a> , optional	IPv6 Netzwerkadresskonfiguration		
	ipv6		
	<b>address_configurations</b> Array of any of <a href="#">NetworkAddressConfigurationIPv6Static</a> , <a href="#">NetworkAddressConfigurationIPv6DHCP</a> or <a href="#">NetworkAddressConfigurationIPv6Auto</a> , optional	NetworkAddressConfigurationIPv6Static[]	
		<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für

Body	JSON-Objekt		
			die Adresse benutzt wird.
		<b>address</b> <u>NetworkInterfaceAddressIPv4</u> (string), <b>required</b>	IPv6 Netzwerkadresse in CIDR-Notation
		<b>gateway</b> <u>NetworkAddressIPv4</u> (string), optional	Standard Gateway für abgehenden Verkehr
		NetworkAddressConfigurationIPv6DHCP[]	
		<b>method</b> string, one of [static, dhcp, auto], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
		NetworkAddressConfigurationIPv6Auto[]	
		<b>method</b> string, one of [static, dhcp, auto], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.

### Beispiele

Entferne alle IPv6 Adresskonfigurationen

```
{
  "ipv6": {
    "address_configurations": []
  }
}
```

Ersetze alle existierenden IPv4 Konfigurationen durch DHCP

```
{
  "ipv4": {
    "address_configurations": [
      {
        "method": "dhcp"
      }
    ]
  }
}
```

Lege statische und dynamische IPv4 Konfiguration fest

```
{
  "ipv4": {
    "address_configurations": [
      {
        "method": "dhcp"
      },
      {
        "method": "static",
        "address": "192.168.0.100/24"
      }
    ]
  }
}
```

Antwort

Code	Body	JSON-Objekt			
200 400 404 423	Properties (object)				
	<b>data</b> Array of <a href="#">NetworkInterfaceAddressConfigurationState</a> , <b>required</b>	NetworkInterfaceAddressConfigurationState[]			
		<b>ipv4</b> <a href="#">NetworkInterfaceAddressFamilyStateIPv4</a> , optional	IPv4 Netzwerkadress-konfiguration		
			ipv4		
			<b>address_configurations</b> Array of any of <a href="#">NetworkAddressConfigurationIPv4Static</a> or <a href="#">NetworkAddressConfigurationIPv4DHCP</a> , optional	NetworkAddressConfigurationIPv4Static[]	
				<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
				<b>address</b> <a href="#">NetworkInterfaceAddressIPv4</a> (string), <b>required</b>	IPv6 Netzwerkadresse in CIDR-Notation
				<b>gateway</b> <a href="#">NetworkAddressIPv4</a> (string), optional	Standard Gateway für abgehenden Verkehr
				NetworkAddressConfigurationIPv4DHCP[]	
				<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
		<b>ipv6</b> <a href="#">NetworkInterfaceAddressFamilyStateIPv6</a> , optional	IPv6 Netzwerkadress-konfiguration		
			ipv6		
			<b>address_configurations</b> Array of any of <a href="#">NetworkAddressConfigurationIPv6Static</a> , <a href="#">NetworkAddressConfigurationIPv6DHCP</a> or <a href="#">NetworkAddressConfigurationIPv6Auto</a> , optional	NetworkAddressConfigurationIPv6Static[]	
				<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
				<b>address</b> <a href="#">NetworkInterfaceAddressIPv4</a> (string), <b>required</b>	IPv6 Netzwerkadresse in CIDR-Notation
				<b>gateway</b> <a href="#">NetworkAddressIPv4</a> (string), optional	Standard Gateway für abgehenden Verkehr
				NetworkAddressConfigurationIPv6DHCP[]	
				<b>method</b>	Konfigurationsmethode,

Code	Body	JSON-Objekt			
				string, one of [static, dhcp, auto], <b>required</b>	welche für die Adresse benutzt wird.
				NetworkAddressConfigurationIPv6Auto[]	
				<b>method</b> string, one of [static, dhcp, auto], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
	<b>errors</b> Array of Error, <b>required</b>	Error[]			
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode		
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte		
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung		
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.validation LPLC.not_found.collection.item				

## 1.2.6 Peripheriegeräte

### 1.2.6.1 Tastaturinformationen ausgeben

**GET** / peripherals / keypad

Gebe grundlegende Informationen über die Tastatur aus.

Antwort

Code	Body	JSON-Objekt	
200	Properties (KeypadInformation)		
	<b>data</b> <u>KeypadInformation</u> , <b>required</b>	Beschreibe den aktuellen Status der Tastatur, als auch Zugriff zu Visualisierungsdaten.	
		KeypadInformation	
		<b>locked</b> boolean, <b>required</b>	Boolean-Kennzeichen, welches Status der Tastatursperre angibt (true -> gesperrt, false -> entsperrt). Alle Tastatureingaben werden ignoriert solange die Sperre aktiv ist.
		<b>clear_matcher_before_teach</b> boolean, <b>required</b>	Das Boolean-Kennzeichen steuert ob mehrere Farben für eine Farbgruppe durch Tastatur-basierte Lernvorgänge gespeichert werden können. Ein Wert aus true impliziert, dass ein Lernvorgang alle existierenden Farben aus der momentan ausgewählten Farbgruppe entfernt, bevor neue Farben hinzugefügt werden. Mit einem Wert von false, werden vorher existierende Farben nicht gelöscht bevor eine neue hinzugefügt wird.
		<b>visualization_url</b> any of string or null, optional, read-only	Der Visualisierungsquellenstandort, kann für das Bereitstellen einer virtuellen Tastaturschnittstelle genutzt werden. Seine URL kann mit einem Schema (z.B. http oder https) für eine volle URL mit Hostnamen starten oder sie startet mit einem Schrägstrich, hinweisend auf einen Weg, welcher von dem Gerät selbst zur Verfügung gestellt wird. Dieses Attribut kann nicht geändert werden.
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung

### 1.2.6.2 Tastatur modifizieren

**PUT** / peripherals / keypad

Ändere Grundzustand der Tastatur.

Anfrage

Body	JSON-Objekt	
Properties (KeypadInformation)		
	<b>locked</b> boolean, required	Boolean-Kennzeichen, welches Status der Tastatursperre angibt (true -> gesperrt, false -> entsperrt). Alle Tastatureingaben werden ignoriert solange die Sperre aktiv ist.
	<b>clear_matcher_before_teach</b> boolean, required	Das Boolean-Kennzeichen steuert ob mehrere Farben für eine Farbgruppe durch Tastatur-basierte Lernvorgänge gespeichert werden können. Ein Wert aus true impliziert, dass ein Lernvorgang alle existierenden Farben aus der momentan ausgewählten Farbgruppe entfernt, bevor neue Farben hinzugefügt werden. Mit einem Wert von false, werden vorher existierende Farben nicht gelöscht bevor eine neue hinzugefügt wird.
	<b>visualization_url</b> any of string or null, optional, read-only	Der Visualisierungsquellenstandort, kann für das Bereitstellen einer virtuellen Tastaturschnittstelle genutzt werden. Seine URL kann mit einem Schema (z.B. <i>http</i> oder <i>https</i> ) für eine volle URL mit Hostnamen starten oder sie startet mit einem Schrägstrich, hinweisend auf einen Weg, welcher von dem Gerät selbst zur Verfügung gestellt wird. Dieses Attribut kann nicht geändert werden.

### Beispiele

```
{
  "locked": true,
  "clear_matcher_before_teach": false,
  "visualization_url": "/media/keypad-image.svg"
}
```

Antwort

Code	Body	JSON-Objekt	
200 400	Properties (KeypadInformation)		
	<b>data</b> <a href="#">KeypadInformation</a> , required	Beschreibe den aktuellen Status der Tastatur, als auch Zugriff zu Visualisierungsdaten.	
		KeypadInformation	
		<b>locked</b> boolean, required	Boolean-Kennzeichen, welches Status der Tastatursperre angibt (true -> gesperrt, false -> entsperrt). Alle Tastatureingaben werden ignoriert solange die Sperre aktiv ist.
		<b>clear_matcher_before_teach</b> boolean, required	Das Boolean-Kennzeichen steuert ob mehrere Farben für eine Farbgruppe durch Tasta-

			tur-basierte Lernvorgänge gespeichert werden können. Ein Wert aus true impliziert, dass ein Lernvorgang alle existierenden Farben aus der momentan ausgewählten Farbgruppe entfernt, bevor neue Farben hinzugefügt werden. Mit einem Wert von false, werden vorher existierende Farben nicht gelöscht bevor eine neue hinzugefügt wird.
		<b>visualization_url</b> any of string or null, optional, read-only	Der Visualisierungsquellenstandort, kann für das Bereitstellen einer virtuellen Tastaturschnittstelle genutzt werden. Seine URL kann mit einem Schema (z.B. http oder https) für eine volle URL mit Hostnamen starten oder sie startet mit einem Schrägstrich, hinweisend auf einen Weg, welcher von dem Gerät selbst zur Verfügung gestellt wird. Dieses Attribut kann nicht geändert werden.
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.validation		

### 1.2.6.3 Benutzerinteraktionen der Tastatur auslesen

**GET** / peripherals / keypad/events

Auflistung der Tastaturfunktionen.

Diese Sammlung dient als Ringpuffer, sodass ältere Ereignisse entfernt werden sobald neue Ereignisse hinzugefügt werden.

Antwort

Code	Body	JSON-Objekt	
200	Properties (object)		
	<b>events</b> Array of <a href="#">KeypadEvent</a> , <b>required</b>	KeypadEvent[]	
		<b>source</b> string, <b>required</b>	Die übliche Quelle der Ereignisse ist <i>inputs</i> .
		<b>name</b> <a href="#">KeypadEventInput</a> (string), <b>required</b>	Name einer Tastatureingabe (Knopf), welcher Ereignisse auslösen könnte.
		<b>event</b> <a href="#">KeypadEventName</a> (string), <b>required</b>	Eingangisperipherie kann verschiedene Ereignisse auslösen.
		<b>timestamp</b> integer, minimum: 0, <b>required</b>	Der Zeitstempel ist in Millisekunden gegeben und sollte monoton steigen.

### 1.2.6.4 Auflistung verfügbarer Eingabelemente

**GET** / peripherals / keypad / inputs

Auflistung der verfügbaren Eingabelemente.

Jedes Tastatureingabelement stellt einen physischen Knopf auf der Tastatur dar.

Antwort

Code	Body	JSON-Objekt			
200	Properties (object)				
	<b>data</b> object, required	data			
		<b>inputs</b> Array of <a href="#">KeypadInputButton</a> , required	KeypadInputButton[]		
			<b>name</b> <a href="#">KeypadEventInput</a> (string), required	Name einer Tastatureingabe (Knopf), welcher Ereignisse auslösen könnte.	
			<b>capabilities</b> Array of object, required	object[]	
				<b>name</b> <a href="#">KeypadEventName</a> (string), required	Eingangspерipherie kann verschiedene Ereignisse auslösen.
				<b>url</b> string, required	Die Ereignisse können extern durch Abgabe einer <b>POST</b> -Anfrage entgegen dieser Quelle ausgelöst werden.
		<b>errors</b> Array of Error, required	Error[]		
			<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode	
			<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte	
			<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung	

### 1.2.6.5 Benutzerinteraktion auf der Tastatur simulieren

**POST** / peripherals / keypad / inputs / {name} / {event}

Simuliert einen Knopfdruck durch externes Auslösen der gegebenen Ereignisse für die Eingabe.

Siehe die Sammlung an Tastatureingaben für eine Liste oder verfügbare URLs zum Auslösen von Ereignissen.

Anfrage

Path Variables
<b>name</b> String, required
<b>event</b> String, required

Antwort

Code	
204 404	Die leere Antwort weist auf Erfolg hin
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.resource.unspecified LPLC.resource.invalid LPLC.illegal_request

1.2.6.6 **Ausgabekonfiguration abrufen**

GET / peripherals / outputs

Antwort

Code	Body	JSON-Objekt	
200	Properties (object)		
	<b>data</b> <a href="#">SwitchingOutputs</a> , required	Elektrische Ausgabezeilen können externe Aktoren in verschiedene elektische Zustände führen.	
		SwitchingOutputs	
		<b>output_driver</b> <a href="#">SwitchingOutputDriver</a> (string), required	Der Ausgabetreiber definiert das elektrische Verhalten der Schaltausgänge. Die unterstützten Ausgabetreiber können über /api/sensor/capabilities ausgelesen werden.
		<b>count</b> integer, required	Anzahl verfügbarer Ausgabezeilen
	<b>errors</b> Array of Error, required	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung

1.2.6.7 **Ausgabekonfiguration anpassen**

PUT / peripherals/ outputs

Anfrage

Body	JSON-Objekt
Properties (SwitchingOutputsWritable)	
<b>output_driver</b> <a href="#">SwitchingOutputDriver</a> (string), required	Der Ausgabetreiber definiert das elektrische Verhalten der Schaltausgänge. Die unterstützten Ausgabetreiber können über /api/sensor/capabilities ausgelesen werden.

Antwort

Code	Body	JSON-Objekt	
200 400	Properties (object)		
	<b>data</b> <a href="#">SwitchingOutputs</a> , required	Elektrische Ausgabezeilen können externe Aktoren in verschiedene elektische Zustände führen.	
		SwitchingOutputs	

		<b>output_driver</b> <u>SwitchingOutputDriver</u> (string), <b>required</b>	Der Ausgabetreiber definiert das elektrische Verhalten der Schaltausgänge. Die unterstützten Ausgabetreiber können über <code>/api/sensor/capabilities</code> ausgelesen werden.
		<b>count</b> integer, <b>required</b>	Anzahl verfügbarer Ausgabezeilen
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.validation		

### 1.2.6.8 Aktuelle RS232 Schnittstellenkonfiguration abrufen

**GET** / peripherals / rs232

Antwort

Code	Body	JSON-Objekt	
200 400	Properties (object)		
	<b>data</b> <u>InterfaceRS232</u> , <b>required</b>	InterfaceRS232	
		<b>protocol</b> any of <u>SerialModbusSettings</u> or <u>SerialElizaSettings</u> , <b>required</b>	SerialModbusSettings
			<b>type</b> string, one of [none, eliza, modbus] , default: eliza, <b>required</b>
			<b>slave_id</b> any of number or null, <b>required</b>
			<b>frame_format</b> string, one of [rtu, ascii] , default: rtu, <b>required</b>
			SerialElizaSettings
			<b>type</b> string, one of [none, eliza, modbus] , default: eliza, <b>required</b>
		<b>baud_rate</b> number, one of [9600, 19200, 115200], <b>required</b>	
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung

### 1.2.6.9 RS232 Schnittstellenkonfiguration anpassen

**PUT** / peripherals / rs232

colorSENSOR CFO

Anfrage

Body	JSON-Objekt	
Properties (InterfaceRS232)		
<b>protocol</b> any of <a href="#">SerialModbusSettings</a> or <a href="#">SerialElizaSettings</a> , <b>required</b>	InterfaceRS232	
	<b>protocol</b> any of <a href="#">SerialModbusSettings</a> or <a href="#">SerialElizaSettings</a> , <b>required</b>	SerialModbusSettings
		<b>type</b> string, one of [none, eliza, modbus] , default: eliza, <b>required</b>
		<b>slave_id</b> any of number or null, <b>required</b>
		<b>frame_format</b> string, one of [rtu, ascii] , default: rtu, <b>required</b>
		SerialElizaSettings
		<b>type</b> string, one of [none, eliza, modbus] , default: eliza, <b>required</b>
	<b>baud_rate</b> number, one of [9600, 19200, 115200], <b>required</b>	

Antwort

Code	Body	JSON-Objekt	
200 400	Properties (object)		
	<b>data</b> <a href="#">InterfaceRS232</a> , <b>required</b>	InterfaceRS232	
		<b>protocol</b> any of <a href="#">SerialModbusSettings</a> or <a href="#">SerialElizaSettings</a> , <b>required</b>	SerialModbusSettings
			<b>type</b> string, one of [none, eliza, modbus] , default: eliza, <b>required</b>
			<b>slave_id</b> any of number or null, <b>required</b>
			<b>frame_format</b> string, one of [rtu, ascii] , default: rtu, <b>required</b>
			SerialElizaSettings
			<b>type</b> string, one of [none, eliza, modbus] , default: eliza, <b>required</b>
		<b>baud_rate</b> number, one of [9600, 19200, 115200], <b>required</b>	
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlercodes ausgeben</b>		

	LPLC.validation		
--	-----------------	--	--

### 1.2.6.10 Triggerereigniss-Statistik abrufen

**GET** / peripherals / trigger-sources

Antwort

Code	Body	JSON-Objekt		
200	Properties (object)			
	<b>data</b> <u>TriggerSourcesStatus</u> , <b>required</b>	Der Sensor hat eine Anzahl von Eingabezeilen welche als Auslösequelle genutzt werden können. Die Ereigniszähler werden periodisch (näherungsweise jede Sekunde) aktualisiert.		
		TriggerSourcesStatus		
		<b>trigger_sources</b> Array of object, <b>required</b>	object[]	
			<b>name</b> string, <b>required</b>	
			<b>event_counters</b> Object, <b>required</b>	event_counters
				<b>edge_falling</b> Number, <b>required</b>
				<b>edge_rising</b> Number, <b>required</b>
				<b>level_low</b> Number, <b>required</b>
		<b>errors</b> Array of Error, <b>required</b>	Error[]	
			<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
			<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
			<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung

### 1.2.6.11 Aktuelle USB-Schnittstellenkonfiguration abrufen

**GET** / peripherals / usb

Antwort

Code	Body	JSON-Objekt		
200	Properties (object)			
	<b>data</b> <u>InterfaceUSB</u> , <b>required</b>	InterfaceUSB		
		<b>protocol</b> Any of <u>SerialModbusSettings</u> or <u>SerialElizaSettings</u> , <b>required</b>	SerialModbusSettings	
			<b>type</b> string, one of [none, eliza, modbus] , default: eliza, <b>required</b>	
			<b>slave_id</b> any of number or null, <b>required</b>	
			<b>frame_format</b> string, one of [rtu, ascii] , default: rtu, <b>required</b>	
			SerialElizaSettings	
			<b>type</b>	

			string, one of [none, eliza, modbus] , default: eliza, <b>required</b>
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung

### 1.2.6.12 USB-Schnittstellenkonfiguration anpassen

**PUT** / peripherals / usb

#### Anfrage

Body	JSON-Objekt
Properties (InterfaceUSB)	
<b>protocol</b> any of <a href="#">SerialModbusSettings</a> or <a href="#">SerialElizaSettings</a> , <b>required</b>	SerialModbusSettings
	<b>type</b> string, one of [none, eliza, modbus] , default: eliza, <b>required</b>
	<b>slave_id</b> any of number or null, <b>required</b>
	<b>frame_format</b> string, one of [rtu, ascii] , default: rtu, <b>required</b>
	SerialElizaSettings
	<b>type</b> string, one of [none, eliza, modbus] , default: eliza, <b>required</b>

#### Antwort

Code	Body	JSON-Objekt	
200 400	Properties (object)		
	<b>data</b> <a href="#">InterfaceUSB</a> , <b>required</b>	InterfaceUSB	
		<b>protocol</b> Any of <a href="#">SerialModbusSettings</a> or <a href="#">SerialElizaSettings</a> , <b>required</b>	SerialModbusSettings
			<b>type</b> string, one of [none, eliza, modbus] , default: eliza, <b>required</b>
			<b>slave_id</b> any of number or null, <b>required</b>
			<b>frame_format</b> string, one of [rtu, ascii] , default: rtu, <b>required</b>
			SerialElizaSettings
			<b>type</b> string, one of [none, eliza, modbus] , default: eliza, <b>required</b>
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte

	<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.validation	

### 1.2.7 Aktionen

Der Sensor kann so programmiert werden, dass er auf spezifische externe oder interne Ereignisse reagiert. Die verfügbaren Aktionen können entweder durch Auslöseeingabezeilen oder durch API-Anfragen ausgelöst werden. Dies erlaubt benutzerdefiniertes Verhalten, z.B. Einlernen von Farben durch eine externe Taste.

Dieser Endpunkt stellt Details für alle verfügbaren Aktionen bereit. Zusätzlich können Aktionen als einmaliger Vorgang ausgeführt werden.

! Siehe *Trigger Aktionen*, wenn Sie Auslöseeingabezeilenereignisse mit spezifischen Aktionen für wiederholende Vorgänge verbinden wollen.

#### 1.2.7.1 Aktionsbeschreibungen

Jede Aktion akzeptiert ein verschiedenes Set aus optionalen oder vorgeschriebenen Argumenten. Sie sind im `argument` Feld jeder Aktion in der Sammlung zusammengefasst. Die detaillierte Beschreibung ihrer Bedeutung und die Spezifikation jedes Verhaltens einer Aktion, sind unten beschrieben.

#### 1.2.7.2 Aktion „enable\_switching\_output“

Das Verhalten der Schaltausgänge verändert sich signifikant während diese Aktion für mindestens ein *Auslöseeigniss* konfiguriert ist. Siehe im Einleitungskapitel über *Schaltausgänge, Auslöser und Haltezeiteinstellungen* für eine detaillierte Spezifikation der verschiedenen Verhaltensweisen.

Argumente: keins

#### 1.2.7.3 Aktion „teach\_single“

Tastet eine neue Farbe (Detectable) ab, sobald die Aktion ausgeführt wird.

Argumente:

- `matcher_id` (optional, default: null): die UUID einer Farbgruppe (Matcher) oder null. Im Fall von null wird die neue Farbe (Detectable) einer Farbgruppe (Matcher), basierend auf `matcher_output_patter` zugewiesen. Im Fall einer ID, welche nicht zu einer bestehenden Farbgruppe (Matcher) gehört, wird eine Farbgruppe (Matcher) mit dieser ID erstellt und der neuen Farbe (Detectable) zugewiesen.
- `matcher_output_pattern` (optional, default: null): Ausgangsmuster der Zielfarbgruppe. Dieses Feld wird nur dann beachtet, wenn die `matcher_id` null ist. Im Fall, dass das `matcher_output_pattern` null sein sollte, wird eine neue Farbgruppe (Matcher) erstellt, wann immer die Aktion ausgeführt wird. Anderenfalls ist das `matcher_output_patern` ein Lexikon, welches ein `states` Feld beinhaltet. Dieses `states` Feld ist soll eine Liste boolesche Ausgabezustände beinhalten. Wenn eine Farbgruppe (Matcher) mit diesem Ausgangsmuster bereits existiert, dann wird die neue Farbe (Detectable) dieser Farbgruppe (Matcher) hinzugefügt. Wenn keine Farbgruppe (Matcher) mit solch einem Ausgangsmuster existiert, dann wird eine neue Farbgruppe (Matcher) für dieses Ausgangsmuster erstellt, bevor ihr die neue Farbe (Detectable) hinzugefügt wird.
- `remove_matcher_detectables_before` (optional, default: true): der boolesche Wert bestimmt ob die Farbe (Detectable), gehörend zur ausgewählten Farbgruppe (Matcher), kurz vor dem Hinzufügen der neuen Farbe (Detectable) entfernt werden soll.

#### 1.2.7.4 Aktion „keylock“

Steuert den Zustand der Tastatursperre.

Ein typischer Ansatz wäre, die steigende Flanke einer Auslösereingabe mit der Aktion, mit dem Parameter `locked true` zu verbinden und die fallende Flanke mit derselben Auslösereingabe mit `false`.

Argumente:

- `locked (required)`: der boolesche Wert bestimmt den gewollten Soll-Zustand der Tastatursperre. Der `true` Wert sperrt die Tastatur. Der `false` Wert entsperrt die Tastatur.

### 1.2.7.5 Aktion „run\_autogain“

Starte eine Autogain-Prozedur.

Höchstwahrscheinlich sollte danach die Aktion „remove\_all\_detectables“, da die Farbwerte, durch die veränderten Abtasteinstellungen möglicherweise nicht mehr genau sind.

Argumente: keine

### 1.2.7.6 Aktion „remove\_all\_detectables“

Löscht die Farbsammlung.

### 1.2.7.7 Aktion „remove\_all\_matchers“

Lösche die Farbgruppensammlung. Dies entfernt auch alle Farben.

Argumente: keine

### 1.2.7.8 Aktionen auflisten

**GET** / actions

Liest eine Liste verfügbarer Aktionen aus.

Antwort

Code	Body	JSON-Objekt		
200	Properties (object)			
	<b>data</b> object, <b>required</b>	data		
		<b>actions</b> Array of <u>Action</u> , <b>required</b>	Action []	
			<b>name</b> string, <b>required</b>	Eindeutiger Name der Aktion.
			<b>arguments</b> object, <b>required</b>	arguments
	<b>errors</b> Array of Error, <b>required</b>	Error[]		
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode	
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte	
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung	

### 1.2.7.9 Aktion abrufen

**GET** / actions / {itemId}

Gibt eine einzige Aktion aus.

Anfrage

<b>Path Variables</b>
<b>itemId</b> String, required

Antwort

Code	Body	JSON-Objekt	
200	Properties (object)		
	<b>data</b> Action, required	Der Sensor erlaubt die Verbindung von Ereignissen mit Aktionen. Aktionen können mit den Sensoroperationen oder den vom Sensor gehandhabten Informationen (z.B. die Liste der gespeicherten Farben (Detectables)) zusammenhängen. Die Aktionen bestehen aus einem eindeutigen Namen und einem Set optionaler Argumente. Die Liste verfügbarer Aktionen und ihre möglichen Argumente können über /api/actions ausgelesen werden.	
		Action	
		<b>name</b> string, required	Eindeutiger Name der Aktion
		<b>arguments</b> object, required	Arguments
	<b>errors</b> Array of Error, required	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item		

1.2.7.10 Software-ausgelöste Aktionen

**POST** / actions / {itemId} / execute

Führe die gegebene Aktion einmal durch. Dies bewerkstelligt Zugang zu allen Vorgängen, welche mit den Trigger Ereignissen verbunden werden können.

Zusätzliche Parameter (falls von der Aktion benötigt) können im Body der Anfrage zur Verfügung gestellt werden. Zum Beispiel die *Tastensperre* Aktion kann durch das Anbieten eines Lexikons, welches den Schlüssel *locked/verschlossen* mit dem gewollten booleschen Soll-Zustand beinhaltet, ausgeführt werden. Siehe das *Argumente* Feld in jeder Aktion in der Sammlung oben.

Der Inhalt der Antwort hängt von der spezifischen Aktion ab, welche ausgeführt wurde.

Die Beispielaktionen mit ihren jeweiligen Parametern und Antworten, werden möglicherweise nicht von allen Sensoren unterstützt. Die *Actions* Sammlung beinhaltet die maßgebliche Liste aller unterstützten Aktionen für jeden Sensor.

Anfrage

<b>Path Variables</b>
<b>itemId</b> String, required

Body	JSON-Objekt		
Properties (AnyAction (union))			

Body	JSON-Objekt			
<b>ActionEnableSwitchingOutput</b> <a href="#">ActionEnableSwitchingOutput</a> , optional	Wende das <i>Ausgangsmuster</i> der aktuell erkannten Farbgruppe auf die Schaltausgänge des Sensors an.			
	ActionEnableSwitchingOutput			
	<b>name</b> string, <b>required</b>	Eindeutiger Name der Aktion		
	<b>arguments</b> object, <b>required</b>	arguments		
<b>ActionTeachDetectable</b> <a href="#">ActionTeachDetectable</a> , optional	Füge die aktuell abgetastete Farbe als eine Farbe (Detectable), der ausgewählten Farbgruppe hinzu.			
	ActionTeachDetectable			
	<b>arguments</b> object, <b>required</b>	arguments		
		<b>matcher_id</b> UUID (string), pattern: <code>^[a-f0-9-]+\$, optional, read-only</code>	Die neue Farbe (Detectable) ist der Farbgruppe (matcher) zugewiesen, welche von dieser UUID erkannt wurde. Im Falle, dass diese Farbgruppen-UUID (und „matcher_output_pattern“) undefiniert sein sollten, wird eine neue Farbgruppe (Matcher) erstellt.	
		<b>matcher_output_pattern:</b> object, optional	Muster der Schaltausgänge, welches beim Auswählen der Zielfarbgruppe für die neue Farbe (Detectable) benutzt wird. Eine passende Farbgruppe (Matcher) wird erstellt, wenn keine Farbgruppe (Matcher) mit dem bestimmten Muster gefunden wurde. Dieses Feld wird ignoriert, wenn die „matcher_id“ nicht null ist. Wenn kein Muster definiert ist (und keine „matcher_id“), dann wird eine neue Farbgruppe (Matcher) erstellt, wann immer eine dazugehörige Aktion ausgeführt wird.	
			matcher_output_pattern:	
			<b>states</b> Array of any of boolean or null, <b>required</b>	Liste der True / False-Werte, die den aktuellen Status der Schaltausgänge beschreiben
			<b>remove_matcher_detectables_before</b> boolean , default: true, optional	Entferne alle Farben (Detectables), welche zur konfigurierten Farbgruppe (Matcher) gehören, bevor die neuen Farbe (Detectable) beigefügt wird.
	<b>name</b> string, <b>required</b>	Eindeutiger Name der Aktion		

Body	JSON-Objekt			
<b>ActionKeyLock</b> <a href="#">ActionKeyLock</a> , optional	Ändere den <i>verriegelten</i> Zustand der Tastatur. Dies erlaubt oder verweigert lokalen Zugang auf den Sensor durch die Tastatur.			
	ActionKeyLock			
	<b>arguments</b> object, <b>required</b>	arguments		
		<b>locked</b> boolean, <b>required</b>	Soll-Zustand der Tastatursperre.	
	<b>name</b> string, <b>required</b>	Eindeutiger Name der Aktion		
<b>ActionRunAutogain</b> <a href="#">ActionRunAutogain</a> , optional	Starte eine automatische Anpassung des optischen Sensor-Setups. Siehe <a href="#">/api/sensor/detection-profiles/current/autogain</a> für Details.			
	ActionRunAutogain			
	<b>name</b> string, <b>required</b>	Eindeutiger Name der Aktion		
	<b>arguments</b> object, <b>required</b>	arguments		
<b>ActionRemoveAllDetectables</b> <a href="#">ActionRemoveAllDetectables</a> , optional	Entferne alle gespeicherten Farben (Detectables), gehörend zu jeglicher Farbgruppe (Matcher).			
	ActionRemoveAllDetectables			
	<b>name</b> string, <b>required</b>	Eindeutiger Name der Aktion		
	<b>arguments</b> object, <b>required</b>	arguments		
<b>ActionRemoveAllMatchers</b> <a href="#">ActionRemoveAllMatchers</a> , optional	Entferne alle gespeicherten Farbgruppen (einschließlich der dazugehörigen Farben (Detectables))			
	ActionRemoveAllMatchers			
	<b>name</b> string, <b>required</b>	Eindeutiger Name der Aktion		
	<b>arguments</b> object, <b>required</b>	arguments		

Antwort

Code	Body	JSON-Objekt			
200 204	Properties (object)				
	<b>data</b> any of <a href="#">ActionResultEnableSwitchingOutput</a> , <a href="#">ActionResultTeachDetectable</a> , <a href="#">ActionResultKeyLock</a> , <a href="#">ActionResultRunAutogain</a> , <a href="#">ActionResultRemoveAllDetectables</a> (string) or <a href="#">ActionResultRemoveAllMatchers</a> (string), <b>required</b>	ActionResultEnableSwitchingOutput			
		<b>uuid</b> <b>UUID</b> (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8		

Code	Body	JSON-Objekt			
		<b>timestamp</b> <a href="#">TimestampBackendUptime</a> (number), <b>required</b>	Der Zeitstempel (in Mikrosekunden angegeben) basiert auf der Betriebszeit des internen Backends des analogen Sensors. Unter bestimmten Bedingungen kann es auf Null zurückgesetzt werden.		
		<b>corrected_color</b> <a href="#">CorrectedColor</a> , <b>required</b>	Darstellung einer Farbe im Farbraum XYZ.		
			<b>values</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	Ort in einem Farbraum	
		<b>transformed_color</b> <a href="#">TransformedColor</a> , <b>required</b>	Eine Farbe, die durch eine Koordinate im Farbraum dargestellt wird. Die Array-Indizes der Eigenschaft values stimmen mit der Reihenfolge der Eigenschaft <code>color-space.axes</code> des aktuell verwendeten Erkennungsprofils überein.		
			<b>values</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	Ort in einem Farbraum	
		<b>representations</b> <a href="#">ColorRepresentations</a> , <b>required</b>	Vorberechnete visuelle Darstellungen einer zum Rendern geeigneten Farbe		
			<b>RGB</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	RGB-Farbarray, das die Achsen r, g und b in dieser Reihenfolge darstellt. Werte sind Gleitkommazahlen zwischen 0 und 1.	
		<b>inputs</b> <a href="#">InputsState</a> , <b>required</b>	Der Status aller Eingaben während eines bestimmten Zeitraums wird durch eine Liste möglicher Ereignisse in Kombination mit einem Booleschen Wert angegeben, der angibt, ob das angegebene Ereignis innerhalb des Zeitraums aufgetreten ist.		
			// <b>boolean</b> , <b>required</b>	Der boolesche Wert gibt an, ob das benannte Eingabeereignis in der letzten Periode aufgetreten ist.	
		<b>detection</b> <a href="#">ColorMatchingResult</a> , <b>required</b>	Nach jeder Abtastperiode wird der abgerufene Farbwert mit den gespeicherten Farben (DetectablesFarbpositionen) verglichen. Farben (Detectables) werden ignoriert, wenn die Toleranzform der entsprechenden Farbgruppe (Matcher) die aktuelle Stichprobe nicht umfasst. Schließlich wird die geeignetste Farbe (Detectable) als Gewinner der Farbanpassungsoperation ausgewählt. Die ent-		

Code	Body	JSON-Objekt			
			sprechende Farbgruppe (Matcher) bestimmt den Zustand des Sensors für die Dauer der nächsten Abtastperiode.		
			detection		
			<b>matcher</b> any of <u>UUID</u> (string) or null, optional, Deprecated	Veraltet: Verwenden Sie stattdessen "selected_matcher_id"	
			<b>chosen_matcher_id</b> any of <u>UUID</u> (string) or null, <b>required</b>	eindeutige Kennung des ausgewählten Matchers	
			<b>distances</b> Array of any of number or null, <b>required</b>	Abstand zwischen der Farbposition der Probe und der nächstgelegenen Farbposition des ausgewählten Matchers entlang der drei Achsen des Farbraums. Das Array enthält drei Nullwerte, wenn für das aktuelle Farbmuster kein geeigneter Matcher gefunden wurde.	
			<b>output_pattern</b> <u>CurrentSwitchingOutputs-State</u> , <b>required</b>	Derzeit aktiver Status der Schaltausgänge. Beachten Sie, dass dies von den angegebenen Ausgangszuständen des aktuell besten Matchers abweichen kann, da Einstellungen wie ausgelöste Eingabe oder Haltezeit den Aktualisierungsprozess für die Schaltausgänge beeinflussen.	
				<b>states</b> Array of any of boolean or null, <b>required</b>	Liste der True / False-Werte, die den aktuellen Status der Schaltausgänge beschreiben
		<b>signal_level</b> number, <b>required</b>	Der Signalpegel zeigt die Verwendung des internen ADC-Abtastbereichs an		
		ActionResultTeachDetectable			
		<b>uuid</b> <u>UUID</u> (string), pattern: ^ [a-f0-9-]+\$, <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8		

Code	Body	JSON-Objekt			
		<b>alias</b> <u>Alias</u> (integer), <b>required</b> , read-only	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.		
		<b>matcher_id</b> <u>UUID</u> (string), pattern: ^[a-f0-9-]+\$, <b>required</b> , read-only	Referenz zu der Farbgruppe (Matcher), welche diese Farbe (Detectable) enthält.		
		<b>color</b> <u>TransformedColor</u> , <b>required</b>	Eine Farbe, welche durch Koordinaten im Farbraum angegeben wird. Die Array-Anzeichen des Besitzes des values entsprechen der Reihenfolge des Besitzes des <code>color-space.axes</code> des aktuellen Erkennungsprofils.		
			color		
			<b>values</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	Ort in einem Farbraum	
		<b>representations</b> ColorRepresentations, optional, read-only	Vorberechnete visuelle Darstellung einer geeigneten Farbe zur Wiedergabe		
		ActionResultKeyLock			
		<b>locked</b> boolean, <b>required</b>	Neuer Zustand der Tastatursperre.		
		ActionResultRunAutogain			
		<b>level</b> Number, default: 0.8, minimum: 0.01, maximum: 1, optional	Soll-Wert für die Autogain-Prozedur		
		<b>minimum_sample_rate</b> SampleRate (number), minimum: 0.02, optional	Gewünschte Abtastrate (der Standard ist die aktuelle Abtastrate)		
		<b>enable_internal_emitter</b> Boolean, default: true, optional	Steuert die Kraft der internen Lichtquelle		
		<b>enable_ambient_light_compensation</b> Boolean, default: true, optional	Steuert das Umgebungslichtkompensationsverfahren. Diese Einstellung ist nur dann relevant, wenn <code>enable_internal_emitter</code> true gesetzt ist. Die Umgebungslichtkompensation führt zu einem gepulsten Gebrauch des internen Lichtemitters. Stichproben werden gesammelt für wechselnde Leucht und Dunkelphasen. Dies erlaubt die Berechnung einer Farbstichprobe des Ziels ausschließlich aller optischen Schnittstellen externer Lichtquellen. Sie sollten Umgebungslichtkompensation nicht ausschalten, solange der optische Pfad perfekt isoliert ist. Anderenfalls beeinträchtigt externes Licht		

Code	Body	JSON-Objekt			
			zwangsläufig die Farbstichprobenerhebung.		
		<b>averages</b> AverageSampleCount (integer), minimum: 1, optional	Anzahl zu mittelnder vorheriger Stichproben für jedes Stichprobenergebnis.		
	<b>errors</b> Array of Error, required	Error[]			
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode		
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte		
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung		

### 1.2.8 Standards

Sammlung von Standards und Einstellungen für spezifische Aufgaben. Neben den benutzerdefinierten Werten, gibt die API ebenso Werkseinstellungen zurück. Standards werden implizit während bestimmter Aktionen, wie der Erstellung von Farbgruppen (Matchers) oder beim Ausführen bestimmter Verhaltensweisen, angewandt.

Client-Anwendungen können diesen API Endpunkt nutzen, um Einstellungen zu speichern, welche unabhängig von ihrer aktuellen Sitzung oder dem Client selbst sind. Typ- und Gültigkeitsprüfungen sind verantwortlich für die Client-Anwendung.

Seien Sie sich im Klaren, dass Standards während ihrer Laufzeit (wie der Erstellung von Farbgruppen (Matchers) angewandt werden können und somit unterbricht ein invalider Standardwert die Anwendung an einem späteren Zeitpunkt. Beachten Sie die unten stehenden Hinweise, um solche Probleme beim Bedienen den Standards der API zu verhindern.

- Standardwerte unterliegen keiner Validierung, werden aber wie Rohdaten behandelt. Eine ungültige `hold_time` (negative, string anstatt einer Zahl) für eine Farbgruppe (Matcher) wird wirksam während der Erstellung einer neuen Farbgruppe (Matcher) und nur wenn die Anfrage zum Erstellen einer neuen Farbgruppe (Matcher) kein `hold_time` Feld beinhaltet. In solch einem Fall, würde die Erstellung einer Farbgruppe (Matcher) scheitern. Es ist daher von zentraler Bedeutung, die Standardwerte richtig zu validieren.
- Die Felder `object_type` und `key` sind von Ihnen zu wählen. Dies erlaubt Anwendungen Sitzungs- und Client-unabhängige Daten (z.B. ein Schnittstellenmotiv, Farbschema, usw.) zu speichern. Um Namenskollisionen mit internen Standard-Feldern zu verhindern, sollten Sie den Feldern `object_type` oder `key`, ein `x-` voranstellen (z.B. `x-theme`, anstatt `theme`). Die API wird intern niemals Felder benutzen, die mit `x-` anfangen.
- Die API beschließt Standards mit den folgenden Schritten. Anwendungen sollten das selbe Verhalten umsetzen, beim Beschließen von Standardwerten:
  1. Prüfe ob ein Element in der `defaults` Sammlung zu `object_type` und `key` passt
  2. Falls ja: benutze diesen Wert
  3. Falls nein: benutze den Wert aus `factory_defaults`

#### 1.2.8.1 Standardwertepaare auflisten

**GET** / defaults

Gibt zwei Sammlungen aus *Standardwertepaare* Objekten aus. `defaults` beinhaltet alle Benutzerstandards und `factory_defaults` alle, welche Teil der Werkseinstellungen sind. Letzteres kann nicht verändert werden, aber Benutzerstandards haben vor Werkseinstellungen Vorrang.

Antwort

Code	Body	JSON-Objekt		
200	<b>Properties (object)</b>			
	<b>data</b> object, <b>required</b>	data		
		<b>defaults</b> Array of <a href="#">DefaultsMapValue</a> , <b>required</b>	DefaultsMapValue[]	
			<b>uuid</b> <a href="#">UUID</a> (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
			<b>object_type</b> string, <b>required</b> , read-only	Name des Objekts das für den Standard gedacht ist
			<b>key</b> string, <b>required</b> , read-only	Name der Objekteigenschaft
			<b>value</b> any, <b>required</b>	Aktueller Standardwert für die Objekteigenschaft
		<b>factory_defaults</b> Array of <a href="#">DefaultsMapValue</a> , <b>required</b>	DefaultsMapValue[]	
			<b>uuid</b> <a href="#">UUID</a> (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
			<b>object_type</b> string, <b>required</b> , read-only	Name des Objekts das für den Standard gedacht ist
			<b>key</b> string, <b>required</b> , read-only	Name der Objekteigenschaft
			<b>value</b> any, <b>required</b>	Aktueller Standardwert für die Objekteigenschaft
	<b>errors</b> Array of Error, <b>required</b>	Error[]		
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode	
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte	
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung	

### 1.2.8.2 Standardwertepaare erstellen

**POST** / defaults

Alle gültigen Attribute für eine PUT-Anfrage eines Standardobjekts sind erlaubt. Die Attribute `object_type`, `key` und `value` werden benötigt. Die API stellt sicher, dass nur eine Kombination aus `object_type` und `key`, zur selben Zeit auftritt. Eine **POST**-Anfrage, erhöht dadurch nicht zwangsweise die Anzahl an Elementen in der Sammlung.

Anfrage

Body	JSON-Objekt
<b>Properties (DefaultsMapValue)</b>	
<b>uuid</b> <a href="#">UUID</a> (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
<b>object_type</b> string, <b>required</b> , read-only	Name des Objekts das für den Standard gedacht ist
<b>key</b> string, <b>required</b> , read-only	Name der Objekteigenschaft
<b>value</b> any, <b>required</b>	Aktueller Standardwert für die Objekteigenschaft

### Beispiele

Farbgruppe (Matcher): Toleranz

```
{
  "uuid": "a7bd36b3-e9c1-4f60-8d7e-cf47634a28b1",
  "object_type": "matcher",
  "key": "tolerance",
  "value": {
    "shape": "sphere",
    "limits": {
      "radius": 4
    }
  }
}
```

Farbgruppe (Matcher): Haltezeit

```
{
  "uuid": "55b35901-1ea6-4b3d-864a-60af15a9b0c5",
  "object_type": "matcher",
  "key": "hold_time",
  "value": 0
}
```

Farbgruppe (Matcher): Setze Ausgabe, nach Haltezeit-Verfall zurück

```
{
  "uuid": "9ba8a7a4-7fa5-4bfc-8883-98d7b6084e91",
  "object_type": "matcher",
  "key": "reset_output_after_hold_time_expired",
  "value": false
}
```

Autogain: Anzahl an Stichproben, welche zur Mittelwertbildung genutzt wurden

```
{
  "uuid": "eeb46031-10e5-4f13-901a-c7eb16aa0cf9",
  "object_type": "autogain",
  "key": "averages",
  "value": 0
}
```

Antwort

Code	Body	JSON-Objekt	
200 400	<b>Properties (object)</b>		
	<b>data</b> DefaultsMapValue, required	DefaultsMapValue	
		<b>uuid</b> UUID (string), pattern: ^[a-f0-9-]+\$, required, read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
		<b>object_type</b> string, required, read-only	Name des Objekts das für den Standard gedacht ist
		<b>key</b> string, required, read-only	Name der Objekteigenschaft
		<b>value</b> any, required	Aktueller Standardwert für die Objekteigenschaft
	<b>errors</b> Array of Error, required	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b>	Visuell lesbare Fehlerbeschreibung

		String, optional	
	<b>Könnte folgende Fehlercodes ausgeben</b>		
	LPLC.validation.collection_size_exceeded		

### 1.2.8.3 Mehrere oder alle Standardwertpaare entfernen

**DELETE** / defaults

Entferne eine Auswahl an Standardwertpaaren, entweder basierend auf einem gegebenem Filterkriterium (falls unterstützt für diese Auswahl) oder entferne alle Standardwertpaare von der Sammlung.

Alle Lösch-Anfragen resultieren in einer leeren Erfolgsrückmeldung (204). Dies ist sogar gültig für eine ungefilterte **DELETE**-Anfrage entgegen einer leeren Sammlung oder für eine gefilterte **DELETE**-Anfrage entgegen einer Sammlung ohne Farbgruppen (Matchers) die dem Filter entsprechen.

Antwort

<b>Code</b>	
204	Die leere Antwort weist auf Erfolg hin

### 1.2.8.4 Standardwertpaare anpassen

**PUT** / defaults / {itemId}

Ändere den Standardwert. Die Felder `uuid`, `object_type` und `key` sind unveränderlich.

Anfrage

<b>PathVariables</b>		
<b>itemId</b> String, required		
<b>Body</b>	<b>JSON-Objekt</b>	
Properties (object)		
<b>data</b> DefaultsMapValue, required	DefaultsMapValue	
	<b>uuid</b> UUID (string), pattern: <code>^[a-f0-9-]+\$, required, read-only</code>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
	<b>object_type</b> string, required, read-only	Name des Objekts das für den Standard gedacht ist
	<b>key</b> string, required, read-only	Name der Objekteigenschaft
	<b>value</b> any, required	Aktueller Standardwert für die Objekteigenschaft

### Beispiele

Farbgruppe (Matcher): Toleranz

```
{
  "uuid": "a7bd36b3-e9c1-4f60-8d7e-cf47634a28b1",
  "object_type": "matcher",
  "key": "tolerance",
  "value": {
    "shape": "sphere",
    "limits": {
      "radius": 4
    }
  }
}
```

Farbgruppe (Matcher): Haltezeit

```
{
  "uuid": "55b35901-1ea6-4b3d-864a-60af15a9b0c5",
  "object_type": "matcher",
  "key": "hold_time",
  "value": 0
}
```

Farbgruppe (Matcher): Setze Ausgabe, nach Haltezeit-Verfall zurück

```
{
  "uuid": "9ba8a7a4-7fa5-4bfc-8883-98d7b6084e91",
  "object_type": "matcher",
  "key": "reset_output_after_hold_time_expired",
  "value": false
}
```

Autogain: Anzahl an Stichproben, welche zur Mittelwertbildung genutzt wurden

```
{
  "uuid": "eeb46031-10e5-4f13-901a-c7eb16aa0cf9",
  "object_type": "autogain",
  "key": "averages",
  "value": 0
}
```

Antwort

Code	Body	JSON-Objekt	
200 400 404	<b>Properties (object)</b>		
	<b>data</b> <a href="#">DefaultsMapValue</a> , <b>required</b>	DefaultsMapValue	
		<b>uuid</b> <a href="#">UUID</a> (string), pattern: <code>^[a-f0-9-]+\$, <b>required</b>, read-only</code>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
		<b>object_type</b> string, <b>required</b> , read-only	Name des Objekts das für den Standard gedacht ist
		<b>key</b> string, <b>required</b> , read-only	Name der Objekteigenschaft
		<b>value</b> any, <b>required</b>	Aktueller Standardwert für die Objekteigenschaft
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item		

1.2.8.5 Standardwertepaar löschen

**DELETE** / defaults / {itemId}

Löscht ein einziges Standardwertepaar.

Anfrage

<b>PathVariables</b>
<b>itemId</b> String, required

Antwort

<b>Code</b>	
204	Die leere Antwort weist auf Erfolg hin
	<b>Könnte folgende Fehlercodes zurückgeben</b> LPLC.not_found.collection.item

1.2.8.6 Standardwertepaar abrufen

**GET** / defaults /{itemId}

Gibt ein einziges Standardwertepaar aus.

Anfrage

<b>PathVariables</b>
<b>itemId</b> String, required

Antwort

Code	Body	JSON-Objekt	
200	<b>Properties (object)</b>		
	<b>data</b> DefaultsMapValue, required	DefaultsMapValue	
		<b>uuid</b> UUID (string), pattern: ^[a-f0-9-]+\$, required, read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
		<b>object_type</b> string, required, read-only	Name des Objekts das für den Standard gedacht ist
		<b>key</b> string, required, read-only	Name der Objekteigenschaft
		<b>value</b> any, required	Aktueller Standardwert für die Objekteigenschaft
	<b>errors</b> Array of Error, required	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item		

1.2.9 Firmware

Die Firmware wird auf dem Gerät gespeichert und steuert all dessen Aspekte. Sie kann aktualisiert werden und sicher auf die Werkseinstellungen zurückgesetzt werden.

1.2.9.1 Firmware-Informationen abrufen

**GET** / firmware

Gibt Informationen über die aktuell laufende Firmware aus.

Antwort

Code	Body	JSON-Objekt	
200	Properties (object)		
	<b>data</b> <u>FirmwareInformation</u> , <b>required</b>	Informationen, welche eine Firmware-Version beschreiben.	
		<b>FirmwareInformation</b>	
		<b>id</b> <u>FirmwareBuildId</u> (string), pattern: <code>^[a-f0-9]+\$</code> , <b>required</b>	Eindeutige ID des aktuell laufenden Firmware-Images
		<b>channel</b> <u>ReleaseChannel</u> (string), one of [stable, feature] , default: stable, <b>required</b>	Beschreibt die Art der Publikation Veröffentlichungen auf dem stable Kanal, gelten generell als gut getestet und werden für die Nutzung in der Produktion empfohlen. Veröffentlichungen auf dem feature Kanal, fügen neue Funktionen hinzu aber wurden nicht so oft wie stable Veröffentlichungen getestet. Feature Veröffentlichungen können, aber sollten nur mit reichlicher Überlegung in der Produktion genutzt werden.
		<b>created_on</b> <u>Timestamp</u> (string), <b>required</b>	Zeit in der diese Firmware erstellt wurde
		<b>name</b> string, <b>required</b>	Visuell lesbarer Name dieser Veröffentlichung
		<b>notes</b> string, <b>required</b>	Versionshinweise formatiert als Absatz
		<b>version</b> <u>FirmwareVersion</u> (string), <b>required</b>	Version einer Firmware
		<b>works_with</b> Array of string, <b>required</b>	Kompatible Gerätemodelle (siehe <code>model_key</code> in <code>/api/device</code> )
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung

### 1.2.9.2 Firmware-Image-Uploads

**GET** / firmware / images

Gibt eine Liste aller aktiven unvollständigen oder vollständigen Firmware-Uploads zurück.

Antwort

Code	Body	JSON-Objekt	
200	Properties (object)		
	<b>data</b> Array of <u>FirmwareImageUpload</u> , <b>required</b>	FirmwareImageUpload []	
		<b>uuid</b> <u>UUID</u> (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	Eindeutige ID eines Firmware-Uploads
		<b>build_id</b> <u>HashDigest</u> (string), pattern: <code>^[a-f0-9]+\$</code> , <b>required</b>	Eindeutige ID des aktuell laufenden Firmware-Images
		<b>status</b>	Aktueller Status des Firmware-Uploads

Code	Body	JSON-Objekt	
		string, one of [incomplete, complete, invalid_signature, processing_failure, malformed_content, device_mismatch], <b>required</b>	unvollständig  Die Anzahl der empfangenden Bytes ist niedriger als die Anzahl der Bytes, welche vollständig angekündigt wurden der Firmware-Upload ist vollständig und neue Firmware kann angewandt werden invalid_signature die Firmware-Prüfsumme passte nicht zum erwarteten Wert processing_failure ein interner, undefinierter Fehler trat, während der Verarbeitung der Firmware auf malformed_content das hochgeladene Firmware-Image nutzt ein unerwartetes Format oder es fehlen essentielle Informationen device_mismatch das Firmware-Image kann auf diesem Gerät nicht angewandt werden
		<b>uploaded_size</b> integer, minimum: 0, <b>required</b>	Anzahl hochgeladener Bytes
		<b>expected_size</b> integer, minimum: 1, <b>required</b>	erwartete Gesamtanzahl an Bytes für das Firmware-Image
		<b>max_chunk_size</b> integer, minimum: 1, <b>required</b>	maximale Größe für einen hochladbaren Datensatz auf dem Gerät
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung

### 1.2.9.3 Firmware hochladen

**POST** / firmware / images

Lade eine neue Firmware für eine Aktualisierung in einen separaten Teil oder als eine einzige formularbasierte Upload-Datei hoch.

### 1.2.9.4 Firmware-Upload-Methoden

Zwei verschiedene Ansätze, zum Hochladen eines Firmware-Images sind verfügbar. Der Client wählt die gewollte Methode, durch das Nutzen des assoziierten Anfrageformats aus:

- \* `multipart` format: Upload the complete image in a single request.
- \* `JSON` body: Successively upload single blocks of the firmware image.

#### Dateibasierter Upload („multipart“):

Der dateibasierte Ansatz ist einfach zu nutzen und für die meisten Situationen vorzuziehen. Es benötigt die Nutzung des multipart/form-data Anfrageformats. Die Anfrage übermittelt der API, die volle Firmware-Image-Datei. Die Antwort der API auf diese Anfrage, wird übermittelt, sobald alle verbundenen Vorgänge abgeschlossen sind.

Das Anfrageformat multipart/form-data ist üblich für dateibasierte HTML-Formen. Somit ist es auch möglich diese Firmware-Upload Methode, auch ohne irgendeinen Client-seitigen Code, mit einer einfachen HTML-Form zu nutzen.

#### Datensatz-basierter Upload (JSON Body):

Die Datensatz-basierte Anfrage, benötigt mehr Aufwand auf der Client-Seite. Diese Anfrage könnte hilfreich sein, wenn Sie fortgeschrittene Ablaufsteuerung oder Statusanzeigen, während des Firmware-Uploads erreichen wollen. Nutzen Sie einen JSON-formatierten Anfrage-Body, wenn Sie diese Methode nutzen wollen.

Die ursprüngliche **POST**-Anfrage erstellt und gibt eine Firmware-Upload Einheit aus (`FirmwareImageUpload`). Sie können ihre UUID zum Hochladen der Datensätze des Firmware-Images, via folgender **POST**-Anfrage, an `/api/firmware/images/UPLOAD_UUID/upload` nutzen. Der Firmware-Upload kann durch eine **POST**-Anfrage an `/api/firmware/images/UPLOAD_UUID/apply` abgeschlossen und angewandt werden.

### 1.2.9.5 Fehlerbehandlung

Im Falle eines unbehebbar Fehlern, gibt die API frühestmöglich einen 400 (Bad Request) Statuscode zurück. Wenn der `apply` Parameter positiv gesetzt wurde, dann wird der Statuscode 424 (Failed Dependency) sein.

#### Anfrage

Body	multipart/form-data	JSON-Objekt
Properties(object)		
<b>firmware_file</b> <code>FirmwareImageFile</code> (file), <b>required</b>	Das aktuelle binäre Firmware-Image. Beachten Sie, dass ein <code>filename</code> (mit einem willkürlichen Wert) versorgt sein muss (Fachsprachlich: Der <code>Content-Disposition</code> Kopf dieses Anfrageteils, braucht ein <code>name</code> und ein <code>filename</code> Feld).	
<b>apply</b> integer, one of [0, 1], default: 0, <b>required</b>	Ob die Firmware angewendet werden soll, sobald sie empfangen und validiert worden ist. Wenn dieses Feld auf 1 gesetzt ist, wird die Firmware sofort angewendet, andernfalls gibt die API die Firmware-Details zurück, damit die Anwendung eine separate Anfrage zur Anwendung der Firmware senden kann.	

#### Antwort

Code	Body	JSON-Objekt	
200 400 424	Properties (object)		
	<b>data</b> <code>FirmwareImageUpload</code> , <b>required</b>	Ein komplett oder teilweise hochgeladenes Firmware-Image, welches für die Aktualisierung der Firmware genutzt wird.	
		<code>FirmwareImageUpload []</code>	
		<b>uuid</b> <code>UUID</code> (string), pattern: <code>^[a-f0-9-]+\$, required</code> , read-only	Eindeutige ID eines Firmware-Uploads
		<b>build_id</b> <code>HashDigest</code> (string), pattern: <code>^[a-f0-9]+\$, required</code>	Eindeutige ID des aktuell laufenden Firmware-Images
		<b>status</b> string, one of [incomplete, complete, invalid_signature, processing_failure, malformed_content, device_mismatch], <b>required</b>	Aktueller Status des Firmware-Uploads unvollständig die Anzahl der empfangenden Bytes ist niedriger als die Anzahl der Bytes, welche vollständig angekündigt wurden der Firmware-Upload ist vollständig und neue Firmware kann angewandt werden invalid_signature die Firmware-Prüfsumme passte nicht zum erwarteten Wert processing_failure ein interner, undefinierter Fehler trat, während der Verarbeitung der Firmware auf malformed_content das hochgeladene Firmware-Image nutzt ein unerwartetes Format oder es fehlen essentielle Informationen device_mismatch

Code	Body	JSON-Objekt	
			das Firmware-Image kann auf diesem erät nicht angewandt werden
		<b>uploaded_size</b> integer, minimum: 0, <b>required</b>	Anzahl hochgeladener Bytes
		<b>expected_size</b> integer, minimum: 1, <b>required</b>	erwartete Gesamtanzahl an Bytes für das Firmware-Image
		<b>max_chunk_size</b> integer, minimum: 1, <b>required</b>	maximale Größe für einen hochladbaren Datensatz auf dem Gerät
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.validation.missing_input LPLC.validation.string LPLC.validation.non_negative_integer LPLC.validation.positive_integer LPLC.validation.smaller_integer LPLC.format.malformed.upload LPLC.internal_error		

### 1.2.9.6 Firmware-Image-Datei abrufen

**GET** / firmware / images / {itemId}

Gibt eine einzige Firmware-Image-Datei aus

Anfrage

Path Variables
<b>itemId</b> String, <b>required</b>

Antwort

Code	Body	JSON-Objekt	
200	Properties (object)		
	<b>data</b> <a href="#">FirmwareImageUpload</a> , <b>required</b>	Ein komplett oder teilweise hochgeladenes Firmware-Image, welches für die Aktualisierung der Firmware genutzt wird.	
		FirmwareImageUpload []	
		<b>uuid</b> <a href="#">UUID</a> (string), pattern: ^[a-f0-9-]+\$ <b>, required, read-only</b>	Eindeutige ID eines Firmware-Uploads
		<b>build_id</b> <a href="#">HashDigest</a> (string), pattern: ^[a-f0-9-]+\$ <b>, required</b>	Eindeutige ID des aktuell laufenden Firmware-Images
		<b>status</b> string, one of [incomplete, complete, invalid_signature, processing_failure, malformed_content, device_mismatch], <b>required</b>	Aktueller Status des Firmware-Uploads unvollständig die Anzahl der empfangenden Bytes ist niedriger als die Anzahl der Bytes, welche vollständig angekündigt wurden der Firmware-Upload ist vollständig und neue Firmware kann angewandt werden invalid_signature die Firmware-Prüfsumme passte nicht zum erwarteten Wert processing_failure

Code	Body	JSON-Objekt	
			ein interner, undefinierter Fehler trat, während der Verarbeitung der Firmware auf malformed_content das hochgeladene Firmware-Image nutzt ein unerwartetes Format oder es fehlen essentielle Informationen device_mismatch das Firmware-Image kann auf diesem Gerät nicht angewandt werden
		<b>uploaded_size</b> integer, minimum: 0, <b>required</b>	Anzahl hochgeladener Bytes
		<b>expected_size</b> integer, minimum: 1, <b>required</b>	erwartete Gesamtanzahl an Bytes für das Firmware-Image
		<b>max_chunk_size</b> integer, minimum: 1, <b>required</b>	maximale Größe für einen hochladbaren Datensatz auf dem Gerät
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item		

### 1.2.9.7 Firmware-Image-Datei löschen

**DELETE** / firmware / images / {itemId}

Löscht einen einzigen Firmware-Image-Uplad.

Anfrage

Path Variables
<b>itemId</b> String, <b>required</b>

Antwort

Code	
204	Die leere Antwort weist auf Erfolg hin.
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item

### 1.2.9.8 Wechsel zu neuer Firmware

**POST** / firmware / images / {itemId} / apply

Wendet die Firmware auf dem Gerät an und überschreibt so das aktuelle System-Image, gefolgt von einem Neustart des Geräts.

Anfrage

Path Variables
<b>itemId</b> String, <b>required</b>

Antwort

Code	

204 400 424 500	Die leere Antwort weist auf Erfolg hin.
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.format.malformed.upload LPLC.internal_error

### 1.2.9.9 Datensatz der Firmware hochladen

**POST** / firmware / images /{itemId} / upload

Lädt einen Datensatz (siehe `max_chunk_size`) für dieses Firmware-Image hoch. Die `Content-Range` wird von dem Client gesendet und von dem Server genutzt, um zu bestimmen, wo der Datensatz in das endgültige Image eingefügt wird. Uploads müssen synchron erfolgen, so dass die Startadresse jedes Datensatzes definiert ist als (`last_chunk_end_address + 1`).

Neue Datensätze können hochgeladen werden, solange der Firmwarestatus als `incomplete` gemeldet wird. Alle anderen Status zeigen entweder einen erfolgreichen oder defekten Firmware-Upload an. Im Falle eines dauerhaften Fehlers werden alle nachfolgenden Datensatz-Uploads mit einem HTTP 400 (*Bad Request*) Statuscode abgebrochen.

#### Anfrage

Path Variables	
<b>itemId</b> String, <b>required</b>	
Request Headers	
<b>Content-Range</b> string, pattern: <code>^bytes\s+\d+-\d+/\d+\$</code> , <b>required</b>	Definiert wo der Datensatz in der Firmware-Image-Datei positioniert wird.

#### Antwort

<b>Code</b>	
204 400	Die leere Antwort weist auf Erfolg hin.
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.format.malformed.upload LPLC.header.content_range.conflicts LPLC.header.content_range.invalid LPLC.header.content_range.missing LPLC.upload.missing_chunk LPLC.payload_too_big

### 1.2.9.10 Wiederherstellungs-Firmware-Informationen abrufen

**GET** / firmware / recovery

Gibt Informationen über die aktuelle Wiederherstellungs-Firmware aus.

#### Antwort

Code	Body	JSON-Objekt	
200	Properties (object)		
	<b>data</b> <code>FirmwareInformation</code> , <b>required</b>	Informationen, welche eine Firmware-Version beschreiben.	
		FirmwareInformation	
		<b>id</b> <code>FirmwareBuildId</code> (string), pattern: <code>^[a-f0-9]+\$</code> , <b>required</b>	Eindeutige ID des aktuell laufenden Firmware-Images
		<b>channel</b> <code>ReleaseChannel</code> (string), one of [stable, feature], default: stable, <b>required</b>	Beschreibt die Art der Publikation

Code	Body	JSON-Objekt	
			Veröffentlichungen auf dem <code>stable</code> Kanal, gelten generell als gut getestet und werden für die Nutzung in der Produktion empfohlen. Veröffentlichungen auf dem <code>feature</code> Kanal, fügen neue Funktionen hinzu aber wurden nicht so oft wie <code>stable</code> Veröffentlichungen getestet. Feature Veröffentlichungen können, aber sollten nur mit reichlicher Überlegung in der Produktion genutzt werden.
		<b>created_on</b> Timestamp (string), <b>required</b>	Zeit in der diese Firmware erstellt wurde
		<b>name</b> string, <b>required</b>	Visuell lesbarer Name dieser Veröffentlichung
		<b>notes</b> string, <b>required</b>	Versionshinweise formatiert als Abschlag
		<b>version</b> FirmwareVersion (string), <b>required</b>	Version einer Firmware
		<b>works_with</b> Array of string, <b>required</b>	Kompatible Gerätemodelle (siehe <code>model_key</code> in <code>/api/device</code> )
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung

### 1.2.9.11 Wiederherstellungs-Firmware aktualisieren

**POST** / firmware / recovery / upgrade-from-current

Ersetzt das gespeicherte Wiederherstellungs-Image mit der aktuellen System-Firmware. Dies ist hilfreich, falls Sie das Wiederherstellungs-Image auf eine aktuellere Version aktualisieren wollen.

Das Werks-Image enthält lediglich die eigentliche Firmware. Es speichert nicht die Konfiguration oder Einstellungen der Sensoren.

Der Aktualisierungsvorgang dauert mehrere Minuten.

Antwort

Code	
204 500	Die leere Antwort weist auf Erfolg hin

### 1.2.9.12 Lese Firmware-Einstellungen aus

**GET** / firmware / settings

Gibt die aktuellen Einstellungen bezüglich der Firmware und möglicher Upgrades zurück.

Antwort

Code	Body	JSON-Objekt	
200	Properties (object)		

	<b>data</b> <i>FirmwareSettings, required</i>	Einstellungen in Bezug auf die Firmware des Geräts und Upgrades.	
		FirmwareSettings	
		<b>release_channel</b> <i>ReleaseChannel (string), one of [stable, feature] , default: stable, required</i>	Beschreibt die Art der Publikation Veröffentlichungen auf dem <i>stable</i> Kanal, gelten generell als gut getestet und werden für die Nutzung in der Produktion empfohlen. Veröffentlichungen auf dem <i>feature</i> Kanal, fügen neue Funktionen hinzu aber wurden nicht so oft wie <i>stable</i> Veröffentlichungen getestet. Feature Veröffentlichungen können, aber sollten nur mit reichlicher Überlegung in der Produktion genutzt werden.
	<b>errors</b> <i>Array of Error, required</i>	Error[]	
		<b>code</b> <i>String, optional</i>	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> <i>String, optional</i>	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> <i>String, optional</i>	Visuell lesbare Fehlerbeschreibung

### 1.2.9.13 /firmware/settings

**PUT** / firmware / settings

#### Anfrage

Body	JSON-Objekt
Properties (FirmwareSettings)	
<b>release_channel</b>	Beschreibt die Art der Publikation Veröffentlichungen auf dem <i>stable</i> Kanal, gelten generell als gut getestet und werden für die Nutzung in der Produktion empfohlen. Veröffentlichungen auf dem <i>feature</i> Kanal, fügen neue Funktionen hinzu aber wurden nicht so oft wie <i>stable</i> Veröffentlichungen getestet. Feature Veröffentlichungen können, aber sollten nur mit reichlicher Überlegung in der Produktion genutzt werden.

#### Beispiele

```
{
  "release_channel": "stable"
}
```

#### Antwort

Code	Body	JSON-Objekt	
200 400	Properties (object)		
	<b>data</b> <i>FirmwareSettings, required</i>	Einstellungen in Bezug auf die Firmware des Geräts und Upgrades.	
		FirmwareSettings	
		<b>release_channel</b> <i>ReleaseChannel (string), one of [stable, feature] , default: stable, required</i>	Beschreibt die Art der Publikation Veröffentlichungen auf dem <i>stable</i> Kanal, gelten generell als gut getestet und werden für die Nutzung in der Produktion empfohlen. Veröffentlichungen auf dem <i>feature</i> Kanal, fügen neue Funktionen hinzu aber wurden nicht so oft wie <i>stable</i> Veröffentlichungen getestet. Feature Veröffentlichungen können, aber sollten nur mit reichlicher Überlegung in der Produktion genutzt werden.

			können, aber sollten nur mit reichlicher Überlegung in der Produktion genutzt werden.
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.validation		

### 1.2.9.14 Firmware-Status abrufen

**GET** / firmware / settings

Gibt Informationen über die aktuell laufende Firmware-Version aus.

Antwort

Code	Body	JSON-Objekt	
200	Properties (object)		
	<b>data</b> <u>FirmwareRunningInformation</u> , <b>required</b>	Informationen, die die aktuell laufende Firmware beschreiben.	
		FirmwareRunningInformation	
		<b>build_id</b> <u>FirmwareBuildId</u> (string), pattern: <code>^[a-f0-9]+\$</code> , <b>required</b>	Eindeutige ID des aktuell laufenden Firmware-Images
		<b>source_url</b> any of string or null, optional	Absolute Basis-URL eines Firmware-Repositorys, das für dieses Gerät geeignete Firmware-Abbilder liefert
		<b>version</b> <u>FirmwareVersion</u> (string), <b>required</b>	Version einer Firmware
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung

### 1.2.10 Zugriffskontrolle

Verwaltung des Datenzugriffs und der Einstellungen des Sensors.

#### 1.2.10.1 Benutzer

Benutzer sind die Identitäten, die zum Zugriff auf die API berechtigt sind.

Für die Authentifizierung eines Benutzers bei der Anmeldung ist ein Kennwort erforderlich.

#### 1.2.10.2 Rollen

Rollen beschreiben einen Satz von Berechtigungen. Jeder Benutzer kann mehreren Rollen angehören.

Die Rolle `anonymous` ist besonders: Sie kann Benutzern nicht zugewiesen werden. Stattdessen beschreibt sie das Set aus Berechtigungen, die sowohl für jede nicht authentifizierte als auch für jede au-

thentifizierte Anfrage gewährt werden. Daher kann diese spezielle Rolle als der minimale Satz von Berechtigungen angesehen werden, der für jeden offen steht.

### 1.2.10.3 Auswertung der Zugriffsberechtigungen

Alle Aktionen können standardmäßig ohne Authentifizierung ausgeführt werden, solange kein Benutzerkonto erstellt wurde.

Wenn mindestens ein Benutzer existiert, wird die Zugriffskontrolle von der API angewendet. So werden die Berechtigungen geprüft, bevor eine eingehende Anforderung verarbeitet wird.

Die Authentifizierung erfolgt über das HTTP-Basis-Authentifizierungsschema.

Die Berechtigung für eine bestimmte Aktion (z.B. *Anzeigen* oder *Bearbeiten*), die auf einen bestimmten API-Endpunkt abzielt, wird wie folgt überprüft:

- Welche Art von Aktion wird vom Benutzer angefordert: *Anzeigen* (GET) oder *Bearbeiten* (POST, PUT, DELETE)?
- Zu welchem *Zugriffsbereich* gehört der Ziel-API-Endpunkt (z.B. Netzwerk)?
- Welche Rollen werden dem authentifizierten Benutzer zugewiesen (z.B. *Operator*)?
- Gehört mindestens eine der Berechtigungen einer dieser Rollen zum angeforderten Bereich und enthält die angeforderte *Aktion*?

Die letzte der obigen Fragen entscheidet, ob ein Antrag bearbeitet oder abgelehnt wird.

### 1.2.10.4 HTTP Antworten

Die folgenden zusätzlichen HTTP-Antworten können bei aktivierter Zugriffskontrolle ausgegeben werden:

HTTP 401 Unauthorized:

- wird zurückgegeben, wenn die Anfrage eine Authentifizierung erfordert, aber keine Anmelde-daten angegeben wurden oder die angegebenen Anmeldedaten abgelehnt wurden (z.B. unbekannter Benutzer oder falsches Passwort). Web-Anwendungen, die mit der API verbunden sind, möchten möglicherweise den `X-WWW-Authenticate-Scheme-Disable` Kopf (siehe unten) verwenden, um zu verhindern, dass der Browser des Benutzers diese Fehlerantwort abfängt.

HTTP 403 Forbidden:

- wird zurückgegeben, wenn die angegebenen Anmeldeinformationen gültig waren, der zugeordnete Benutzer die angegebene *Aktion* im *Zielbereich* jedoch nicht anfordern darf.

Der Client kann in jeder Anforderung den HTTP-Kopf `X-WWW-Authenticate-Scheme-Disable` angeben. Es wird erwartet, dass der Inhalt dieses Headers eine durch Kommata getrennte Liste von Authentifizierungsschemata ist (siehe RFC 7235). Diese Authentifizierungsschemata werden von der API nicht als Teil des `WWW-Authenticate` Kopfs in ihrer Antwort beworben. Der folgende Beispiel-Kopf-Inhalt ist geeignet, einen Browser daran zu hindern, authentifizierungsbezogene Antworten zu stören: `-WWW-Authentifizierungs-Schema-Deaktivieren: Basic, Digest.`

### 1.2.10.5 Untersuche Bereiche der Zugriffskontrolle und Aktionen

GET / access

Untersuche die verfügbaren Aspekte der Zugriffskontroll-Einrichtung.

Antwort

Code	Body	JSON-Objekt	

200	Properties (object)		
	<b>data</b> <i>AccessControlFeatures, required</i>	AccessControlFeatures	
		<b>actions</b> Array of <i>AccessAction</i> (string), <b>required</b>	Verfügbare Aktionen, die via Berechtigungen erlaubt oder abgelehnt werden können.
		<b>scopes</b> Array of <i>AccessScope</i> (string), <b>required</b>	Verfügbare Bereiche, auf die mit den verschiedenen Aktionen zugegriffen werden kann.
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung

### 1.2.10.6 In ein Konto einloggen

**POST** / access / login

Dieser Endpunkt kann benutzt werden, um eine Sitzung für den aktuellen Benutzer-Agent zu erstellen oder um Zugangsdaten zu testen.

Wenn der Anrufer gültige Anmeldedaten angegeben hat und den `session_timeout` nicht 0 gesetzt hat, enthält die Antwort einen `Set-Cookie` Kopf, der ein Session-Token enthält, das für die zukünftige Authentifizierung verwendet wird und das automatisch von `XMLHttpRequest` und `fetch` behandelt wird.

Sobald ein Sitzungs-Token an den Benutzer-Agenten ausgegeben wurde, wird bei jeder nachfolgenden Anforderung an einen der API-Endpunkte das Sitzungs-Timeout auf den von `session_timeout` bereitgestellten Wert zurückgesetzt, sofern die Sitzung zu dem Zeitpunkt, zu dem die Anforderung an die API gesendet wurde, noch nicht abgelaufen ist. Wenn also das Sitzungs-Timeout auf 15 Minuten gesetzt wurde und alle 10 Minuten eine Anforderung gestellt wurde, wäre die Sitzung unbegrenzt gültig. Dies gilt nicht, wenn der Endpunkt explizit vom Zurücksetzen des Sitzungs-Timeout ausgenommen ist (wie **GET** /api/access/login).

Zum Testen der Anmeldedaten kann der Client diese zusammen mit `session_timeout`, das auf 0 gesetzt ist, senden. Der Antwortstatuscode zeigt an, ob die Anmeldedaten gültig sind, lässt aber den `Set-Cookie` Kopf weg, so dass das aktuell verwendete Session-Token erhalten bleibt.

**Beachten Sie**, dass die API anfangen kann, die Rate des Endpunkts zu begrenzen, wenn zu viele ungültige Anmeldedaten an sie gesendet wurden. Stellen Sie sicher, dass Sie im Fall einer Antwort mit HTTP Statuscode 429 eine Art Benutzer-Feedback implementieren, wie z.B. die Deaktivierung des Anmeldeformulars und/oder die Anzeige der verbleibenden Zeit bis zu einem erneuten Anmeldeversuch.

#### Anfrage

Body	JSON-Objekt
Properties (object)	
<b>username</b> string, <b>required</b>	Der Name des Nutzers, welcher authentifiziert werden soll
<b>password</b> string, <b>required</b>	Das Passwort des Nutzers, welcher authentifiziert werden soll
<b>session_timeout</b> number, default: 1200, optional	Die Lebensdauer der Sitzung auf der Server-Seite in Sekunden. Die Übergabe von 0 verhindert, dass die API den Set-Cookie Kopf setzt, und ermöglicht die Überprüfung der Anmeldeinformationen, ohne eine neue Sitzung zu erstellen.

Antwort

Code	Body	JSON-Objekt		
200 403 429	Authentifizierung mit den vorgelegten Anmelde­daten war erfolgreich			
	Properties (object)			
	<b>data</b> <a href="#">LoginInformation</a> , <b>required</b>	Beschreibt die aktuell aktiven Anmeldung durch den Benutzer-Agenten		
		LoginInformation		
		<b>logged_in_user</b> any of <a href="#">User</a> or null, <b>required</b>	Der aktuell angemeldete Benutzer. Ist null, wenn die Anmelde­daten zu keinem bekannten Benutzer passen oder abgelaufen sind.	
			User	
			<b>name</b> string, pattern: <code>^[w-]+\$, <b>required</b>, read-only</code>	Eindeutiger Name, der ein Konto identifiziert
			<b>password</b> string, optional	Diesem Konto zugewie­senes Passwort (nur be­schreibbar; wird in Ant­worten nie zurückgege­ben). Beim Anlegen ei­nes neuen Benutzers o­der beim Ändern eines Passworts muss entwe­der ein <code>password</code> oder ein <code>password_hash</code> angegeben werden.
			<b>password_hash</b> <a href="#">HashDigest</a> (string), pat­tern: <code>^[a-f0-9]+\$, op­tional</code>	Diesem Konto zugewie­se­ner Passwort-Hash. Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entwe­der ein <code>password</code> oder ein <code>password_hash</code> angegeben werden.
			<b>roles</b> Array of string, optional	Die einem Konto zuge­weisenen Rollen, defi­nieren den Berechti­gungssatz des Kontos
		<b>session_timeout</b> any of number or null, <b>required</b>	Anzahl der Sekunden, die dieser Sitzung noch verbleiben, bevor sie abläuft. Ist null, wenn die angegebenen Anmelde­daten keiner aktiven Sitzung zugeordnet werden konnten, wenn die Sitzung abgelaufen ist oder wenn der angegebene Authentifizie­rungsmechanismus keine Sitzungen unter­stützt (z.B. HTTP-Au­thentifizierung).	
	<b>errors</b> Array of Error, <b>required</b>	Error[]		
		<b>code</b> String, optional	Maschinenlesbarer spe­zifischer Fehlercode	

Code	Body	JSON-Objekt		
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte	
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung	

### 1.2.10.7 Informationen über die aktuell genutzten Anmeldedaten abrufen

**GET** / access / login

Antwort

Code	Body	JSON-Objekt		
200	Properties (object)			
	<b>data</b> <u>LoginInformation</u> , required	Beschreibt die aktuell aktive Anmeldung durch den Benutzer-Agenten		
		LoginInformation		
		<b>logged_in_user</b> any of <u>User</u> or null, required	Der aktuell angemeldete Benutzer. Ist null, wenn die Anmeldedaten zu keinem bekannten Benutzer passen oder abgelaufen sind.	
			User	
			<b>name</b> string, pattern: <code>^[w-]+\$</code> , required, read-only	Eindeutiger Name, der ein Konto identifiziert
			<b>password</b> string, optional	Diesem Konto zugewiesenes Passwort (nur beschreibbar; wird in Antworten nie zurückgegeben). Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein <code>password</code> oder ein <code>password_hash</code> angegeben werden.
			<b>password_hash</b> <u>HashDigest</u> (string), pattern: <code>^[a-f0-9]+\$</code> , optional	Diesem Konto zugewiesener Passwort-Hash. Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein <code>password</code> oder ein <code>password_hash</code> angegeben werden.
			<b>roles</b> Array of string, optional	Die einem Konto zugewiesenen Rollen, definieren den Berechtigungssatz des Kontos
		<b>session_timeout</b> any of number or null, required	Anzahl der Sekunden, die dieser Sitzung noch verbleiben, bevor sie abläuft. Ist null, wenn die angegebenen Anmeldedaten keiner aktiven Sitzung zugeordnet werden konnten, wenn die Sitzung abgelaufen ist oder wenn der angegebene Authentifizierungsmechanismus keine Sitzungen unterstützt (z.B. HTTP-Authentifizierung).	

Code	Body	JSON-Objekt		
	<b>errors</b> Array of Error, required	Error[]		
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode	
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte	
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung	

### 1.2.10.8 Logout / Alle aktuellen Anmeldeinformationen ungültig machen

**DELETE** / access / login

Antwort

Code	
204 400 403	Die angegebenen Zugangsdaten wurden erfolgreich ungültig gemacht.

### 1.2.10.9 Benutzer anlegen

**POST** / access / users

Erstelle einen neuen Benutzer.

Alle unterstützten Datenattribute im Body der Anfrage sind optional.

Anfrage

Body	JSON-Objekt
Properties (User)	
<b>name</b> string, pattern: ^[\w-]+\$, <b>required</b> , read-only	Eindeutiger Name, der ein Konto identifiziert
<b>password</b> string, optional	Diesem Konto zugewiesenes Passwort (nur beschreibbar; wird in Antworten nie zurückgegeben). Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein password oder ein password_hash angegeben werden.
<b>password_hash</b> HashDigest (string), pattern: ^[a-f0-9]+\$, optional	Diesem Konto zugewiesener Passwort-Hash. Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein password oder ein password_hash angegeben werden.
<b>roles</b> Array of string, optional	Die einem Konto zugewiesenen Rollen, definieren den Berechtigungssatz des Kontos

### Beispiele

```
{
  "name": "alice"
}
```

Antwort

Code	Body	JSON-Objekt	
200 400	Properties (object)		
	<b>data</b> <u>User</u> , <b>required</b>	User	
		<b>name</b> string, pattern: ^[\w-]+\$, <b>required</b> , read-only	Eindeutiger Name, der ein Konto identifiziert

Code	Body	JSON-Objekt	
		<b>password</b> string, optional	Diesem Konto zugewiesenes Passwort (nur beschreibbar; wird in Antworten nie zurückgegeben). Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein password oder ein password_hash angegeben werden.
		<b>password_hash</b> <u>HashDigest</u> (string), pattern: ^[a-f0-9]+\$ , optional	Diesem Konto zugewiesener Passwort-Hash. Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein password oder ein password_hash angegeben werden.
		<b>roles</b> Array of string, optional	Die einem Konto zugewiesenen Rollen, definieren den Berechtigungssatz des Kontos
	<b>errors</b> Array of Error, required	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.validation.collection_size_exceeded		

### 1.2.10.10 Entferne mehrere oder alle Benutzer

**DELETE** / access / users

Entferne eine Auswahl an Benutzern, entweder basierend auf einem gegebenem Filterkriterium (falls unterstützt für diese Auswahl) oder entferne alle Benutzer von der Sammlung.

Alle Lösch-Anfragen resultieren in einer leeren Erfolgsrückmeldung (204). Dies ist sogar gültig für eine ungefilterte **DELETE**-Anfrage entgegen einer leeren Sammlung oder für eine gefilterte **DELETE**-Anfrage entgegen einer Sammlung ohne Benutzer die dem Filter entsprechen.

Antwort

Code	
204	Die leere Antwort weist auf Erfolg hin

### 1.2.10.11 Benutzer auflisten

**GET** / access / users

Gibt eine Liste verfügbarer Benutzer aus.

Antwort

Code	Body	JSON-Objekt	
200	Properties (object)		
	<b>data</b> Object, required	data	
		<b>users</b> Array of <u>User</u> , required	User[]
		<b>name</b> string, pattern: ^[w-]+\$ , required, read-only	Eindeutiger Name, der ein Konto identifiziert

Code	Body	JSON-Objekt		
			<b>password</b> string, optional	Diesem Konto zugewiesenes Passwort (nur beschreibbar; wird in Antworten nie zurückgegeben). Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein <code>password</code> oder ein <code>password_hash</code> angegeben werden.
			<b>password_hash</b> HashDigest (string), pattern: <code>^[a-f0-9]+\$</code> , optional	Diesem Konto zugewiesener Passwort-Hash. Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein <code>password</code> oder ein <code>password_hash</code> angegeben werden.
			<b>roles</b> Array of string, optional	Die einem Konto zugewiesenen Rollen, definieren den Berechtigungssatz des Kontos
	<b>errors</b> Array of Error, <b>required</b>	Error[]		
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode	
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte	
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung	

### 1.2.10.12 Benutzer löschen

**DELETE** / access / users / {name}

Löscht einen einzelnen Benutzer.

Anfrage

Path Variables
<b>name</b> String, <b>required</b>

Antwort

Code	
204	Die leere Antwort weist auf Erfolg hin
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item

### 1.2.10.13 Benutzer anpassen

**PUT** / access / users / {name}

Modifiziert einen einzelnen Benutzer.

Anfrage

Path Variables	
<b>name</b> String, <b>required</b>	
Body	JSON-Objekt
Properties (User)	
<b>name</b> string, pattern: <code>^[w-]+\$</code> , <b>required</b> , read-only	Eindeutiger Name, der ein Konto identifiziert
<b>password</b> string, optional	Diesem Konto zugewiesenes Passwort (nur beschreibbar; wird in Antworten nie zurückgegeben). Beim Anlegen eines neuen Benutzers oder beim

	Ändern eines Passworts muss entweder ein password oder ein password_hash angegeben werden.
<b>password_hash</b> HashDigest (string), pattern: ^[a-f0-9]+\$ , optional	Diesem Konto zugewiesener Passwort-Hash. Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein password oder ein password_hash angegeben werden.
<b>roles</b> Array of string, optional	Die einem Konto zugewiesenen Rollen, definieren den Berechtigungssatz des Kontos

**Beispiele**

```
{
  "name": "alice"
}
```

**Antwort**

Code	Body	JSON-Objekt	
200 400 404	Properties (object)		
	<b>data</b> User, <b>required</b>	User[]	
		<b>name</b> string, pattern: ^[\w-]+\$ , <b>required</b> , read-only	Eindeutiger Name, der ein Konto identifiziert
		<b>password</b> string, optional	Diesem Konto zugewiesenes Passwort (nur beschreibbar; wird in Antworten nie zurückgegeben). Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein password oder ein password_hash angegeben werden.
		<b>password_hash</b> HashDigest (string), pattern: ^[a-f0-9]+\$ , optional	Diesem Konto zugewiesener Passwort-Hash. Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein password oder ein password_hash angegeben werden.
		<b>roles</b> Array of string, optional	Die einem Konto zugewiesenen Rollen, definieren den Berechtigungssatz des Kontos
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item		

**1.2.10.14 Benutzer abrufen**

```
GET / access / users / {name}
```

Gibt einen einzigen Benutzer aus.

**Anfrage**

<b>Path Variables</b>
<b>Name</b> String, <b>required</b>

**Antwort**

Code	Body	JSON-Objekt	
200	Properties (object)		

	<b>data</b> User, <b>required</b>	User[]	
		<b>name</b> string, pattern: <code>^[w-]+\$</code> , <b>required</b> , read-only	Eindeutiger Name, der ein Konto identifiziert
		<b>password</b> string, optional	Diesem Konto zugewiesenes Passwort (nur beschreibbar; wird in Antworten nie zurückgegeben). Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein <code>password</code> oder ein <code>password_hash</code> angegeben werden.
		<b>password_hash</b> <u>HashDigest</u> (string), pattern: <code>^[a-f0-9]+\$</code> , optional	Diesem Konto zugewiesener Passwort-Hash. Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein <code>password</code> oder ein <code>password_hash</code> angegeben werden.
		<b>roles</b> Array of string, optional	Die einem Konto zugewiesenen Rollen, definieren den Berechtigungssatz des Kontos
	<b>errors</b> Array of Error, <b>required</b>	Error[]	
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode
		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung
	<b>Könnte folgende Fehlercodes ausgeben</b> LPLC.not_found.collection.item		

### 1.2.10.15 Zugriffsrollen auflisten

**GET** / access / roles

Gibt eine Liste verfügbarer Zugriffsrollen aus.

Antwort

Code	Body	JSON-Objekt		
200	Properties (object)			
	<b>data</b> object, <b>required</b>	data		
		<b>roles</b> Array of <u>AccessRole</u> , <b>required</b>	AccessRole[]	
			<b>id</b> string, <b>required</b>	
			<b>permissions</b> Array of <u>AccessPermission</u> , <b>required</b>	AccessPermission[]
			<b>scope</b> <u>AccessScope</u> (string), <b>required</b>	Jeder API-Endpoint gehört zu einem Zugriffskontrollbereich. Ein Bereich kombiniert mit einer Reihe von Aktionen bildet eine Berechtigung.
			<b>actions</b> Array of <u>AccessAction</u> (string), <b>required</b>	
	<b>errors</b> Array of Error, <b>required</b>	Error[]		
		<b>code</b> String, optional	Maschinenlesbarer spezifischer Fehlercode	

		<b>mapping</b> String, optional	Eine Referenz zum Parameter welcher den Fehler verursachte		
		<b>message</b> String, optional	Visuell lesbare Fehlerbeschreibung		

## 1.3 Typenbezeichnung

### 1.3.1 AccessAction

Verfügbare Aktionen, die zugelassen oder abgelehnt werden via Berechtigungen.

#### Typinformationen

[AccessAction](#) (string)

#### Beispiele

view

### 1.3.2 Zugriffskontrollfunktionen

#### Eigenschaften

<b>actions</b> Array of <a href="#">AccessAction</a> (string), <b>required</b>	Verfügbare Aktionen, die durch Genehmigungen erlaubt oder verweigert werden können.
<b>scopes</b> Array of <a href="#">AccessScope</a> (string), <b>required</b>	Verfügbare Bereiche, auf die mit den verschiedenen Aktionen zugegriffen werden kann.

#### Beispiel

```
{
  "actions": [
    "view",
    "edit"
  ],
  "scopes": [
    "access",
    "miscellaneous",
    "network",
    "notify",
    "peripherals",
    "sensor",
    "settings",
    "system"
  ]
}
```

### 1.3.3 Zugriffsgenehmigung

Eine Genehmigung beschreibt ein Set erlaubter Aktionen in einem spezifischen Zugriffskontrollbereich.

#### Eigenschaften

<b>scope</b> Array of <a href="#">AccessScope</a> (string), <b>required</b>	Jeder API Endpunkt gehört zu einem Zugriffskontrollbereich. Ein Bereich kombiniert mit einer Anzahl aus Aktionen, bildet eine Genehmigung.
<b>actions</b> Array of <a href="#">AccessAction</a> (string), <b>required</b>	

### 1.3.4 Zugriffsrolle

Eine Rolle beschreibt ein Set aus Genehmigungen. Jeder Benutzer kann mehreren Rollen angehören.

#### Eigenschaften

<b>id</b> string, <b>required</b>		
--------------------------------------	--	--

<b>permissions</b> Array of <a href="#">AccessPermission</a> , <b>required</b>	AccessPermission[]	
	<b>scope</b> <a href="#">AccessScope</a> (string), <b>required</b>	Jeder API Endpunkt gehört zu einem Zugriffskontrollbereich. Ein Bereich kombiniert mit einer Anzahl aus Aktionen, bildet eine Genehmigung.
	<b>actions</b> Array of <a href="#">AccessAction</a> (string), <b>required</b>	

### 1.3.5 Zugriffsbereich

Jeder API Endpunkt gehört zu einem Zugriffskontrollbereich. Ein Bereich kombiniert mit einer Anzahl aus Aktionen, bildet eine Genehmigung.

#### Typinformationen

[AccessScope](#) (string)

#### Beispiele

`network`

### 1.3.6 Aktion

Der Sensor ermöglicht die Verbindung von Ereignissen mit Aktionen. Aktionen können sich auf die Sensorvorgänge oder auf die vom Sensor verarbeiteten Informationen (z.B. die Liste der gespeicherten Farben (Detectables) beziehen.

Die Aktion besteht aus einem eindeutigen Namen und aus einem Set optionaler Argumente.

Die Liste verfügbarer Aktionen und ihre möglichen Argumente können via `/api/actions` ausgelesen werden.

#### Eigenschaften

<b>name</b> string, <b>required</b>	Eindeutiger Name der Aktion
<b>arguments</b> object, <b>required</b>	arguments

### 1.3.7 Aktion Aktivierte Schaltausgang

Wende das `output_pattern` der aktuell erkannten Farbgruppe (Matcher) auf die Schaltausgänge des Sensors an.

#### Eigenschaften

<b>name</b> string, <b>required</b>	Eindeutiger Name der Aktion
<b>arguments</b> object, <b>required</b>	arguments

### 1.3.8 Aktion Tastatursperre

Ändere den `verriegelten` Zustand der Tastatur. Dies erlaubt oder verweigert lokalen Zugriff auf den Sensor durch die Tastatur.

#### Eigenschaften

<b>arguments</b> object, <b>required</b>	arguments
<b>name</b> string, <b>required</b>	Eindeutiger Name der Aktion

#### Beispiel

colorSENSOR CFO

```
{
  "name": "keylock",
  "arguments": {
    "locked": true
  }
}
```

### 1.3.9 Aktion Entferne alle Farben (Detectables)

Entferne alle gespeicherten Farben (Detectables), die zu einer beliebigen Farbgruppe (Matcher) gehören.

#### Eigenschaften

<b>name</b> string, required	Eindeutiger Name der Aktion
<b>arguments</b> object, required	arguments

### 1.3.10 Aktion Entferne alle Farbgruppen (Matchers)

Entferne alle gespeicherten Farbgruppen (Matchers) (einschließlich der dazugehörigen Farben (Detectables)).

#### Eigenschaften

<b>name</b> string, required	Eindeutiger Name der Aktion
<b>arguments</b> object, required	arguments

### 1.3.11 Aktionsergebnis Aktiviere Schaltausgang

Nach jeder Abtastperiode wird, basierend auf der aktuell abgetasteten Farbe und des Inhalts des Farbspeichers (Farbgruppen (Matchers) und Farben (Detectables)), ein Erkennungsergebnis ermittelt.

Zusätzlich zur abgetasteten Farbe beinhaltet das Erkennungsergebnis Übergänge und Ereignisse auf allen Eingangsleitungen während der letzten Abtastperiode sowie den Zustand der Schaltausgänge während der folgenden Abtastperiode.

#### Eigenschaften

<b>uuid</b> <a href="#">UUID</a> (string), pattern: <code>^[a-f0-9-]+\$</code> , required, read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8		
<b>timestamp</b> <a href="#">TimestampBackendUptime</a> (number), required	Der Zeitstempel (in Mikrosekunden angegeben) basiert auf der Betriebszeit des internen Backends des analogen Sensors. Unter bestimmten Bedingungen kann es auf Null zurückgesetzt werden.		
<b>corrected_color</b> <a href="#">CorrectedColor</a> , required	Darstellung einer Farbe im Farbraum XYZ.		
	<b>values</b> Array of number, minimum items: 3, maximum items: 3, required	Ort in einem Farbraum	
<b>transformed_color</b> <a href="#">TransformedColor</a> , required	Eine Farbe, die durch eine Koordinate im Farbraum dargestellt wird. Die Array-Indizes der Eigenschaft <code>values</code> stimmen mit der Reihenfolge der		

	Eigenschaft <code>color-space.axes</code> des aktuell verwendeten Erkennungsprofils überein.		
	<b>values</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	Ort in einem Farbraum	
<b>representations</b> <a href="#">ColorRepresentations</a> , <b>required</b>	Vorberechnete visuelle Darstellungen einer zum Rendern geeigneten Farbe		
	<b>RGB</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	RGB-Farrray, das die Achsen r, g und b in dieser Reihenfolge darstellt. Werte sind Gleitkommazahlen zwischen 0 und 1.	
<b>inputs</b> <a href="#">InputsState</a> , <b>required</b>	Der Status aller Eingaben während eines bestimmten Zeitraums wird durch eine Liste möglicher Ereignisse in Kombination mit einem Booleschen Wert angegeben, der angibt, ob das angegebene Ereignis innerhalb des Zeitraums aufgetreten ist.		
	// <b>boolean</b> , <b>required</b>	Der boolesche Wert gibt an, ob das benannte Eingabeereignis in der letzten Periode aufgetreten ist.	
<b>detection</b> <a href="#">ColorMatchingResult</a> , <b>required</b>	Nach jeder Abtastperiode wird der abgerufene Farbwert mit den gespeicherten detectables (Eingelernten Farben/Farbpositionen) verglichen. Erkennbare Elemente werden ignoriert, wenn die Toleranzform des entsprechenden Matchers (Farbgruppe) die aktuelle Stichprobe nicht umfasst. Schließlich wird die Farbgruppe den am besten übereinstimmenden Farbwerten ausgewählt. Der entsprechende Matcher bestimmt den Zustand des Sensors für die Dauer der nächsten Abtastperiode.		
	<b>matcher</b> any of <a href="#">UUID</a> (string) or null, optional, <b>Deprecated</b>	Veraltet: Verwenden Sie stattdessen "selected_matcher_id"	
	<b>chosen_matcher_id</b> any of <a href="#">UUID</a> (string) or null, <b>required</b>	eindeutige Kennung des ausgewählten Matchers	
	<b>distances</b> Array of any of number or null, <b>required</b>	Abstand zwischen der Farbposition der Probe und der nächstgelegenen Farbposition des ausgewählten Matchers entlang der drei Achsen des Farbraums. Das Array enthält drei Nullwerte, wenn für das aktuelle Farbmuster kein geeigneter Matcher gefunden wurde.	
	<b>output_pattern</b> <a href="#">CurrentSwitchingOutputsState</a> , <b>required</b>	Derzeit aktiver Status der Schaltausgänge. Beachten Sie, dass dies von den angegebenen Ausgangszuständen des aktuell besten Matchers abweichen kann, da Einstel-	

		lungen wie ausgelöste Eingabe oder Haltezeit den Aktualisierungsprozess für die Schaltausgänge beeinflussen.	
		<b>states</b> Array of any of boolean or null, <b>required</b>	Liste der True / False-Werte, die den aktuellen Status der Schaltausgänge beschreiben
<b>signal_level</b> number, <b>required</b>	Der Signalpegel zeigt die Verwendung des internen ADC-Abtastbereichs an		

**Beispiele**

```
{
  "uuid": "4575656f-efe4-4a7d-862c-9660c15cdf4e",
  "timestamp": 12455148861,
  "corrected_color": {
    "values": [
      0.419219434261322,
      0.4271118938922882,
      0.18753691017627716
    ]
  },
  "transformed_color": {
    "values": [
      78.10789489746094,
      5.271166801452637,
      -32.290863037109375
    ]
  },
  "representations": {
    "RGB": [
      0.6569499359485452,
      0.7560762577592035,
      0.9910401649653352
    ]
  },
  "inputs": {
    "trigger_0_level_low": true,
    "trigger_0_edge_falling": false,
    "trigger_0_edge_rising": false,
    "trigger_0_level_high": false
  },
  "detection": {
    "chosen_matcher_id": null,
    "distances": [
      null,
      null,
      null
    ],
    "output_pattern": {
      "states": [
        true,
        true,
        true
      ]
    ]
  },
  "signal_level": 0.7
}
```

### 1.3.12 Aktionsergebnis Tastatursperre

#### Eigenschaften

<b>locked</b> boolean, required	Neuer Zustand der Tastatursperre.
------------------------------------	-----------------------------------

#### Beispiele

```
{
  "locked": true
}
```

### 1.3.13 Aktionsergebnis Entferne alle Farben (Detectables)

Die Antwort ist leer und gibt den HTTP Status 204 aus.

#### Typinformationen

[ActionResultRemoveAllDetectables](#) (string)

### 1.3.14 Aktionsergebnis Entferne alle Farbgruppen (Matchers)

Die Antwort ist leer und gibt den HTTP Status 204 aus.

#### Typinformationen

[ActionResultRemoveAllMatchers](#) (string)

### 1.3.15 Aktionsergebnis Autogain ausführen

Optionale Einstellungen zum Anpassen der Autogain-Prozedur.

#### Eigenschaften

<b>level</b> Number, default: 0.8, minimum: 0.01, maximum: 1, optional	Soll-Wert für die Autogain-Prozedur
<b>minimum_sample_rate</b> SampleRate (number), minimum: 0.02, optional	Gewünschte Abtastrate (der Standard ist die aktuelle Abtastrate)
<b>enable_internal_emitter</b> Boolean, default: true, optional	Steuert die Kraft der internen Lichtquelle
<b>enable_ambient_light_compensation</b> Boolean, default: true, optional	Steuert das Umgebungslichtkompensationsverfahren. Diese Einstellung ist nur dann relevant, wenn enable_internal_emitter true gesetzt ist. Die Umgebungslichtkompensation führt zu einem gepulsten Gebrauch des internen Lichtemitters. Stichproben werden gesammelt für wechselnde Leucht und Dunkelphasen. Dies erlaubt die Berechnung einer Farbstichprobe des Ziels ausschließlich aller optischen Schnittstellen externer Lichtquellen. Sie sollten Umgebungslichtkompensation nicht ausschalten, solange der optische Pfad perfekt isoliert ist. Andernfalls beeinträchtigt externes Licht zwangsläufig die Farbstichprobenerhebung.
<b>averages</b> AverageSampleCount (integer), minimum: 1, optional	Anzahl zu mittelnder vorheriger Stichproben für jedes Stichprobenergebnis.

#### Beispiele

```
{
  "level": 0.7,
  "minimum_sample_rate": 1500,
  "enable_internal_emitter": true,
  "enable_ambient_light_compensation": true
}
colorSENSOR CFO
```

}

### 1.3.16 Aktionsergebnis Farbe (Detectable) einlernen

Eine Farbe (Detectable) vertritt die numerische Position in einem Farbraum. Sie ist verbunden mit einer Farbgruppe (Matcher).

#### Eigenschaften

<b>uuid</b> UUID (string), pattern: <code>^[a-f0-9-]+\$, required, read-only</code>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
<b>alias</b> Alias (integer), <b>required, read-only</b>	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.	
<b>matcher_id</b> UUID (string), pattern: <code>^[a-f0-9-]+\$, required, read-only</code>	Referenz zu der Farbgruppe (Matcher), welche diese Farbe (Detectable) enthält.	
<b>color</b> TransformedColor, <b>required</b>	Eine Farbe, welche durch Koordinaten im Farbraum angegeben wird. Die Array-Anzeichen des Besitzes des <code>values</code> entsprechen der Reihenfolge des Besitzes des <code>color-space.axes</code> des aktuellen Erkennungsprofils.	
	<b>color</b>	
	<b>values</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	Ort in einem Farbraum
<b>representations</b> ColorRepresentations, optional, read-only	Vorberechnete visuelle Darstellung einer geeigneten Farbe zur Wiedergabe	
	<b>representations</b>	
	<b>RGB</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	RGB-Farbararray, das die Achsen r, g und b in dieser Reihenfolge darstellt. Werte sind Gleitkommazahlen zwischen 0 und 1.

#### Beispiele

```
{
  "uuid": "9f968e8a-ad9c-45ce-9beb-a55011856a99",
  "alias": 2,
  "matcher_id": "1c7e9725-8753-4b6c-a0b7-a71d7e915cb5",
  "color": {
    "values": [
      0.476731,
      0.381263,
      0.128475
    ]
  },
  "representations": {
    "RGB": [
      0.396114,
      0.479113,
      0.552308
    ]
  }
}
```

### 1.3.17 Aktion Autogain ausführen

Starte eine automatische Anpassung des optikolen Sensor-Setups. Siehe `/api/sensor/detection-profiles/current/autogain` für Details.

#### Eigenschaften

<b>name</b> string, required	Eindeutiger Name der Aktion
<b>arguments</b> object, required	arguments

### 1.3.18 Aktion Farbe (Detectable) einlernen

Füge die aktuell abgetastete Farbe als eine Farbe (Detectable) der ausgewählten Farbgruppe (Matcher) hinzu.

#### Eigenschaften

<b>arguments</b> object, required	arguments		
	<b>matcher_id</b> UUID (string), pattern: <code>^[a-f0-9-]+\$, optional, read-only</code>	Die neue Farbe (Detectable) ist der Farbgruppe (matcher) zugewiesen, welche von dieser UUID erkannt wurde. Im Falle, dass diese Farbgruppen-UUID (und „matcher_output_pattern“) undefiniert sein sollten, wird eine neue Farbgruppe (Matcher) erstellt.	
	<b>matcher_output_pattern:</b> object, optional	Muster der Schaltausgänge, welches beim Auswählen der Zielfarbgruppe für die neue Farbe (Detectable) benutzt wird. Eine passende Farbgruppe (Matcher) wird erstellt, wenn keine Farbgruppe (Matcher) mit dem bestimmten Muster gefunden wurde. Dieses Feld wird ignoriert, wenn die „matcher_id“ nicht null ist. Wenn kein Muster definiert ist (und keine „matcher_id“), dann wird eine neue Farbgruppe (Matcher) erstellt, wann immer eine dazugehörige Aktion ausgeführt wird.	
		matcher_output_pattern:	
		<b>states</b> Array of any of boolean or null, required	Liste der True / False-Werte, die den aktuellen Status der Schaltausgänge beschreiben
	<b>remove_matcher_detectables_before</b> boolean, default: true, optional	Entferne alle Farben (Detectables), welche zur konfigurierten Farbgruppe (Matcher) gehören, bevor die neuen Farbe (Detectable) beigefügt wird.	
<b>name</b> string, required	Eindeutiger Name der Aktion		

#### Beispiele

```
{
  "name": "teach_single",
  "arguments": {
    "matcher_id": "3f26aff4-8650-42a0-b319-51776c443fbc",
    "remove_matcher_detectables_before": false
  }
}
```

### 1.3.19 Aktionsauslöser

Ein Aktionsauslöser ordnet einem Ereignis ein bestimmtes Set aus Aktionen zu.

Am Ende jeder Abtastperiode werden alle Ereignisse ausgewertet. Danach werden alle entsprechenden Aktionen ausgeführt.

#### Eigenschaften

<b>uuid</b> UUID (string), pattern: <code>^[a-f0-9-]+\$,</code> <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
<b>event</b> TriggerEventName (string), <b>required</b>	Jeder Fallname angeboten von <code>/api/sensor/capabilities</code> (Attribut <code>trigger_sources</code> ) ist erlaubt.	
<b>actions</b> Array of Action, <b>required</b>	Liste der auszuführenden Aktionen nach dem gegebenem Fall	
	Action[]	
	<b>name</b> String, <b>required</b>	Eindeutiger Name der Aktion
	<b>arguments</b> Object, <b>required</b>	arguments

#### Beispiele

```
{
  "uuid": "3f26aff4-8650-42a0-b319-51776c443fbc",
  "event": "trigger_0_edge_falling",
  "actions": [
    {
      "name": "enable_switching_output",
      "arguments": {}
    }
  ]
}
```

### 1.3.20 Alias

Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.

#### Typinformationen

[Alias](#) (integer)

#### Beispiele

4

### 1.3.21 Amplifikationslevel

Das Amplifikationslevel gibt die interne Konfiguration eines Verstärkers an. Der Wert ist nicht dazu gedacht von normalen Nutzern beeinflusst zu werden. Er soll genutzt werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).

#### Typinformationen

[AmplificationLevel](#) (integer)

### 1.3.22 Jede Aktion

## Typinformationen

any of [ActionEnableSwitchingOutput](#), [ActionTeachDetectable](#), [ActionKeyLock](#), [ActionRunAutogain](#), [ActionRemoveAllDetectables](#) or [ActionRemoveAllMatchers](#)

### 1.3.23 Jedes Aktionsergebnis

## Typinformationen

any of [ActionResultEnableSwitchingOutput](#), [ActionResultTeachDetectable](#), [ActionResultKeyLock](#), [ActionResultRunAutogain](#), [ActionResultRemoveAllDetectables](#) (string) or [ActionResultRemoveAllMatchers](#) (string)

### 1.3.24 Autogain-Einstellungen

Optionale Einstellungen zum Anpassen des Autogain-Verfahrens.

## Eigenschaften

<b>level</b> Number, default: 0.8, minimum: 0.01, maximum: 1, optional	Soll-Wert für die Autogain-Prozedur
<b>minimum_sample_rate</b> SampleRate (number), minimum: 0.02, optional	Gewünschte Abtastrate (der Standard ist die aktuelle Abtastrate)
<b>enable_internal_emitter</b> Boolean, default: true, optional	Steuert die Kraft der internen Lichtquelle
<b>enable_ambient_light_compensation</b> Boolean, default: true, optional	Steuert das Umgebungslichtkompensationsverfahren. Diese Einstellung ist nur dann relevant, wenn <code>enable_internal_emitter</code> true gesetzt ist. Die Umgebungslichtkompensation führt zu einem gepulsten Gebrauch des internen Lichtemitters. Stichproben werden gesammelt für wechselnde Leucht- und Dunkelphasen. Dies erlaubt die Berechnung einer Farbstichprobe des Ziels ausschließlich aller optischen Schnittstellen externer Lichtquellen. Sie sollten Umgebungslichtkompensation nicht ausschalten, solange der optische Pfad perfekt isoliert ist. Andernfalls beeinträchtigt externes Licht zwangsläufig die Farbstichprobenerhebung.
<b>averages</b> AverageSampleCount (integer), minimum: 1, optional	Anzahl zu mittelnder vorheriger Stichproben für jedes Stichprobenergebnis.

## Beispiele

```
{
  "level": 0.7,
  "minimum_sample_rate": 1500,
  "enable_internal_emitter": true,
  "enable_ambient_light_compensation": true
}
```

### 1.3.25 Durchschnittliche Stichprobenzahl

Anzahl der vorherigen Stichproben, die für jedes Stichprobenergebnis gemittelt werden soll. Ein gleitender Mittelungsalgorithmus wird auf die Stichproben angewendet.

## Typinformationen

[AverageSampleCount](#) (integer), minimum: 1

### 1.3.26 Grundfarbtoleranz

#### Eigenschaften

<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
<b>limits</b> object, <b>required</b>	limits

### 1.3.27 Serielle Grundeinstellungen

#### Eigenschaften

<b>type</b> string, one of [none, eliza, modbus] , default: eliza, <b>required</b>
---

### 1.3.28 Box-Farbtoleranz

#### Eigenschaften

<b>limits</b> Object, <b>required</b>	limits <b>half_edges</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>
<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .

### 1.3.29 Farbwertkoordinate

Position in einem Farbraum.

#### Typinformationen

Array of number, minimum items: 3, maximum items: 3

### 1.3.30 Farbe (ColorDetectable)

Eine Farbe (Detectable) vertitt die numerische Position in einem Farbraum. Sie ist verbunden mit einer Farbgruppe (Matcher).

#### Eigenschaften

<b>uuid</b> UUID (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
<b>alias</b> Alias (integer), <b>required</b> , read-only	Ein numerischer Wert, welcher benutzt werden kann, um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.	
<b>matcher_id</b> UUID (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	Referenz zu der Farbgruppe (Matcher), welche diese Farbe (Detectable) enthält.	
<b>color</b> TransformedColor, <b>required</b>	Eine Farbe, welche durch Koordinaten im Farbraum angegeben wird. Die Array-Anzeichen des Besitzes des <code>values</code> entsprechen der Reihenfolge des Besitzes des <code>color-space.axes</code> des aktuellen Erkennungsprofils.	
	color	
	<b>values</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	Ort in einem Farbraum
<b>representations</b> ColorRepresentations, optional, read-only	Vorberechnete visuelle Darstellung einer geeigneten Farbe zur Wiedergabe	

	representations	
	<b>RGB</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	RGB-Farbararray, das die Achsen r, g und b in dieser Reihenfolge darstellt. Werte sind Gleitkommazahlen zwischen 0 und 1.

**Beispiele**

```
{
  "uuid": "9f968e8a-ad9c-45ce-9beb-a55011856a99",
  "alias": 2,
  "matcher_id": "1c7e9725-8753-4b6c-a0b7-a71d7e915cb5",
  "color": {
    "values": [
      0.476731,
      0.381263,
      0.128475
    ]
  },
  "representations": {
    "RGB": [
      0.396114,
      0.479113,
      0.552308
    ]
  }
}
```

**1.3.31 Farberkennungsergebnis**

Nach jeder Abtastperiode wird, basierend auf der aktuell abgetasteten Farbe und des Inhalts des Farbspeichers (Farbgruppen (Matchers) und Farben (Detectables)), ein Erkennungsergebnis ermittelt.

Zusätzlich zur abgetasteten Farbe beinhaltet das Erkennungsergebnis Übergänge und Ereignisse auf allen Eingangsleitungen während der letzten Abtastperiode sowie den Zustand der Schaltausgänge während der folgenden Abtastperiode.

**Eigenschaften**

<b>uuid</b> <a href="#">UUID</a> (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8		
<b>timestamp</b> <a href="#">TimestampBackendUptime</a> (number), <b>required</b>	Der Zeitstempel (in Mikrosekunden angegeben) basiert auf der Betriebszeit des internen Backends des analogen Sensors. Unter bestimmten Bedingungen kann es auf Null zurückgesetzt werden.		
<b>corrected_color</b> <a href="#">CorrectedColor</a> , <b>required</b>	Darstellung einer Farbe im Farbraum XYZ.		
	<b>values</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	Ort in einem Farbraum	
<b>transformed_color</b> <a href="#">TransformedColor</a> , <b>required</b>	Eine Farbe, die durch eine Koordinate im Farbraum dargestellt wird. Die Array-Indizes der Eigenschaft <code>values</code> stimmen mit der Reihenfolge der Eigenschaft <code>colorspace.axes</code> des aktuell verwendeten Erkennungsprofils überein.		

	<b>values</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	Ort in einem Farbraum	
<b>representations</b> <a href="#">ColorRepresentations</a> , <b>required</b>	Vorberechnete visuelle Darstellungen einer zum Rendern geeigneten Farbe		
	<b>RGB</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	RGB-Farbararray, das die Achsen r, g und b in dieser Reihenfolge darstellt. Werte sind Gleitkommazahlen zwischen 0 und 1.	
<b>inputs</b> <a href="#">InputsState</a> , <b>required</b>	Der Status aller Eingaben während eines bestimmten Zeitraums wird durch eine Liste möglicher Ereignisse in Kombination mit einem Booleschen Wert angegeben, der angibt, ob das angegebene Ereignis innerhalb des Zeitraums aufgetreten ist.		
	// <b>boolean</b> , <b>required</b>	Der boolesche Wert gibt an, ob das benannte Eingabeereignis in der letzten Periode aufgetreten ist.	
<b>detection</b> <a href="#">ColorMatchingResult</a> , <b>required</b>	Nach jeder Abtastperiode wird der abgerufene Farbwert mit den gespeicherten detectables (Eingelernten Farben/Farbpositionen) verglichen. Erkennbare Elemente werden ignoriert, wenn die Toleranzform des entsprechenden Matchers (Farbgruppe) die aktuelle Stichprobe nicht umfasst. Schließlich wird die Farbgruppe den am besten übereinstimmenden Farbwerten ausgewählt. Der entsprechende Matcher bestimmt den Zustand des Sensors für die Dauer der nächsten Abtastperiode.		
	<b>matcher</b> any of <a href="#">UUID</a> (string) or null, optional, <b>Deprecated</b>	Veraltet: Verwenden Sie stattdessen "selected_matcher_id"	
	<b>chosen_matcher_id</b> any of <a href="#">UUID</a> (string) or null, <b>required</b>	eindeutige Kennung des ausgewählten Matchers	
	<b>distances</b> Array of any of number or null, <b>required</b>	Abstand zwischen der Farbposition der Probe und der nächstgelegenen Farbposition des ausgewählten Matchers entlang der drei Achsen des Farbraums. Das Array enthält drei Nullwerte, wenn für das aktuelle Farbmuster kein geeigneter Matcher gefunden wurde.	
	<b>output_pattern</b> <a href="#">CurrentSwitchingOutputsState</a> , <b>required</b>	Derzeit aktiver Status der Schaltausgänge. Beachten Sie, dass dies von den angegebenen Ausgangszuständen des aktuell besten Matchers abweichen kann, da Einstellungen wie ausgelöste Eingabe oder Haltezeit den Aktualisierungsprozess für die Schaltausgänge beeinflussen.	
		<b>states</b> Array of any of boolean or null, <b>required</b>	Liste der True / False-Werte, die den aktuellen Status der Schaltausgänge beschreiben
<b>signal_level</b> number, <b>required</b>	Der Signalpegel zeigt die Verwendung des internen ADC-Abtastbereichs an		

**Beispiele**

{

colorSENSOR CFO

```
"uuid": "4575656f-efe4-4a7d-862c-9660c15cdf4e",
"timestamp": 12455148861,
"corrected_color": {
  "values": [
    0.419219434261322,
    0.4271118938922882,
    0.18753691017627716
  ]
},
"transformed_color": {
  "values": [
    78.10789489746094,
    5.271166801452637,
    -32.290863037109375
  ]
},
"representations": {
  "RGB": [
    0.6569499359485452,
    0.7560762577592035,
    0.9910401649653352
  ]
},
"inputs": {
  "trigger_0_level_low": true,
  "trigger_0_edge_falling": false,
  "trigger_0_edge_rising": false,
  "trigger_0_level_high": false
},
"detection": {
  "chosen_matcher_id": null,
  "distances": [
    null,
    null,
    null
  ],
  "output_pattern": {
    "states": [
      true,
      true,
      true
    ]
  }
},
"signal_level": 0.7
}
```

### 1.3.32 Farberkennungsergebnisliste

#### Typinformationen

Array of [ColorDetectionResult](#)

### 1.3.33 Farberkennungsergebnis oder Null

#### Typinformationen

Any of [ColorDetectionResult](#) or null

### 1.3.34 Farbgruppe (ColorMatcher)

Eine Farbgruppe vertritt ein bestimmtes Erkennungsergebnis und das gewollte Verhalten des Sensors, wann immer er angetroffen wird.

#### Eigenschaften

<b>uuid</b> UUID (string), pattern: <code>^[a-f0-9-]+\$,</code> <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
<b>alias</b> Alias (integer), <b>required</b> , read-only	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.	
<b>name</b> String, <b>required</b>	Visuell lesbarer Name der Farbgruppe (matcher)	
<b>tolerance</b> Any of InfiniteColorTolerance, SphereColorTolerance, CylinderColorTolerance or BoxColorTolerance, <b>required</b>	Spezifikation einer geometrischen Form und ihrer Dimension in dem aktuellen Farbraum.	
	InfiniteColorTolerance	
	<b>limits</b> Object, <b>required</b>	limits
	<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
	SphereColorTolerance	
	<b>limits</b> Object, <b>required</b>	limits <b>radius</b> Numer, <b>required</b>
	<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
	CylinderColorTolerance	
	<b>limits</b> Object, <b>required</b>	limits <b>radius</b> Number, <b>required</b> <b>half_height</b> Number, <b>required</b>
	<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
	BoxColorTolerance	
	<b>limits</b> Object, <b>required</b>	limits <b>half_edges</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>
	<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
<b>output_pattern</b> WantedSwitchingOutputsState, <b>required</b>	Die Kombination von tristate Werten beschreibt einen logischen Zustand der Schaltausgänge des Sensors. Die Zustände <code>true</code> oder <code>false</code> veranlassen den Ausgang anzusteigen oder zu sinken. Der Zustand <code>null</code> lässt den vorherigen Zustand der Ausgabe unverändert.	
	<b>uuid</b> UUID (string), pattern: <code>^[a-f0-9-]+\$,</code> <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8

	<b>states</b> Array of any of boolean or null, <b>required</b>	Liste von True/False/Null Werten beschreibt den gesuchten Zustand der Schaltausgänge.
<b>hold_time</b> HoldTime (number), maximum: 3153600000, <b>required</b>	Minimum Laufzeit (in Sekunden) der Ausgabekonfiguration einer Farbgruppe, bei Anwendung nach der Erkennung.	
<b>reset_output_after_hold_time_expired</b> Boolean, default: false, <b>required</b>	Steuert ob die Ausgabe zurückgesetzt werden soll, nachdem die Haltezeit überschritten wurde. Dies ist hilfreich, wenn Sie nur durch auslösende Eingaben abfragen und wünschen im Nachhinein die Ausgaben zurückzusetzen.	
<b>signal_color</b> Any of string or null, <b>required</b>	Ein benutzerdefinierter Farbname. Wie und welche Farbe angezeigt wird, wird durch den Kunden bestimmt.	

**Beispiele**

```
{
  "uuid": "9ffaa31f-8011-44f5-bb2a-f91e4be50764",
  "alias": 6,
  "name": "clean bottle cap",
  "tolerance": {
    "limits": {
      "radius": 2,
      "half_height": 4
    },
    "shape": "cylinder"
  },
  "output_pattern": {
    "uuid": "1adc74e2-96ac-4761-b9e6-2d93e02d9244",
    "states": [
      true,
      false,
      false
    ]
  },
  "hold_time": 0,
  "reset_output_after_hold_time_expired": false,
  "signal_color": null
}
```

**1.3.35 Farbanpassungsergebnis**

Nach jeder Abtastperiode wird der abgerufene Farbwert mit den gespeicherten Farben (Detectables-Farbpositionen) verglichen. Farben (Detectables) werden ignoriert, wenn die Toleranzform der entsprechenden Farbgruppe (Matcher) die aktuelle Stichprobe nicht umfasst. Schließlich wird die geeignetste Farbe (Detectable) als Gewinner der Farbanpassungsoperation ausgewählt. Die entsprechende Farbgruppe (Matcher) bestimmt den Zustand des Sensors für die Dauer der nächsten Abtastperiode.

**Eigenschaften**

<b>matcher</b> any of UUID (string) or null, optional, <b>Deprecated</b>	Veraltet: Verwenden Sie stattdessen "selected_matcher_id"	
<b>chosen_matcher_id</b> any of UUID (string) or null, <b>required</b>	eindeutige Kennung des ausgewählten Matchers	
<b>distances</b> Array of any of number or null, <b>required</b>	Abstand zwischen der Farbposition der Probe und der nächstgelegenen Farbposition des ausgewählten Matchers entlang der drei Achsen des Farbraums. Das Array enthält drei Nullwerte, wenn für das aktuelle Farbmuster kein geeigneter Matcher gefunden wurde.	

<b>output_pattern</b> CurrentSwitchingOutputs- State, required	Derzeit aktiver Status der Schaltausgänge. Beachten Sie, dass dies von den angegebenen Ausgangszuständen des aktuell besten Matchers abweichen kann, da Einstellungen wie ausgelöste Eingabe oder Haltezeit den Aktualisierungsprozess für die Schaltausgänge beeinflussen.	
	<b>states</b> Array of any of boolean or null, required	Liste der True / False-Werte, die den aktuellen Status der Schaltausgänge beschreiben

**Beispiele**

Keine Übereinstimmung

```
{
  "chosen_matcher_id": null,
  "distances": [
    null,
    null,
    null
  ],
  "output_pattern": {
    "states": [
      true,
      true,
      true
    ]
  }
}
```

Passende Übereinstimmung

```
{
  "chosen_matcher_id": "4575656f-efe4-4a7d-862c-9660c15cdf4e",
  "distances": [
    1.4,
    0.3,
    null
  ],
  "output_pattern": {
    "states": [
      true,
      false,
      true
    ]
  }
}
```

**1.3.36 Farbdarstellung**

Vorberechnete visuelle Darstellungen einer zum Rendern geeigneten Farbe.

**Eigenschaften**

<b>RGB</b> Array of number, minimum items: 3, maximum items: 3, required	RGB-Farrray, das die Achsen r, g und b in dieser Reihenfolge darstellt. Werte sind Gleitkommazahlen zwischen 0 und 1.
---	---

**Beispiele**

```
{
  colorSENSOR CFO
```

```

"RGB": [
  0.3197475,
  0.754686,
  0.216748
]
}

```

### 1.3.37 Farbraum

Ein Farbraum beschreibt die numerische Umwandlung von Farben unter bestimmten Bedingungen. Verschiedene standardisierte Farbräume sind passend für verschiedene Erkennungsaufgaben.

#### Eigenschaften

<b>name</b> String, <b>required</b>		
<b>space_id</b> ColorspaceID, <b>required</b>	Eindeutiger Name eines Farbraums	
<b>axes</b> Array of ColorspaceAxis, minimum items: 3, maximum items: 3, <b>required</b>	ColorspaceAxis[]	
	<b>id</b> String, <b>required</b>	Eindeutiger Name
	<b>label</b> String, <b>required</b>	Visuell lesbarer Name
	<b>minimum</b> Number, <b>required</b>	Niedrigster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
	<b>maximum</b> Number, <b>required</b>	Höchster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen

#### Beispiele

```

{
  "name": "L*a*b*",
  "space_id": "Lab",
  "axes": [
    {
      "id": "L",
      "label": "L*",
      "minimum": 0,
      "maximum": 100
    },
    {
      "id": "a",
      "label": "a*",
      "minimum": -500,
      "maximum": 500
    },
    {
      "id": "b",
      "label": "b*",
      "minimum": -200,
      "maximum": 200
    }
  ]
}

```

### 1.3.38 Farbraumachsen

#### Eigenschaften

<b>id</b> String, <b>required</b>	Eindeutiger Name
<b>label</b> String, <b>required</b>	Visuell lesbarer Name
<b>minimum</b> Number, <b>required</b>	Niedrigster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
<b>maximum</b> Number, <b>required</b>	Höchster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen

### 1.3.39 Farbraum-ID

Eindeutiger Name eines Farbraums

#### Typinformationen

[ColorspaceID](#) (string)

### 1.3.40 Farbraumtoleranzkarte

Legen Sie die Verwendung der Achsen der einzelnen Farbräume für nicht-triviale Toleranzformen fest. Siehe "colorspace\_tolerance\_maps" unter "/capabilities" für weitere Details.

#### Eigenschaften

<b>colorspace_id</b> ColorspaceID (string), <b>required</b>	Eindeutiger Name eines Farbraums
<b>tolerance_shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .
<b>limits_axes_map</b> Object, <b>required</b>	limits_axes_map
	<b>half_height</b> Array of string, optional
	<b>half_edges</b> Array of string, optional
	<b>radius</b> Array of string, optional

#### Beispiele

```
{
  "colorspace_id": "Lab",
  "tolerance_shape": "cylindrical",
  "limits_axes_map": {
    "half_height": [
      "L"
    ],
    "radius": [
      "a",
      "b"
    ]
  }
}
```

### 1.3.41 Farbtoleranz

Angabe einer geometrischen Form und ihrer Abmessungen in den aktuellen Farbräumen.

#### Typinformationen

Any of [InfiniteColorTolerance](#), [SphereColorTolerance](#), [CylinderColorTolerance](#) or [BoxColorTolerance](#)

#### Beispiele

Unbegrenzt

```
{
  "shape": "infinite",
  "limits": {}
}
```

Kugel

```
{
  "shape": "sphere",
  "limits": {
    "radius": 2
  }
}
```

Zylidner

```
{
  "shape": "cylinder",
  "limits": {
    "radius": 2,
    "half_height": 4
  }
}
```

Box

```
{
  "shape": "box",
  "limits": {
    "half_edges": [
      4,
      2,
      2
    ]
  }
}
```

**1.3.42 Kompensationseinstellungen**

Die Kompensationseinstellungen eines Erkennungsprofils beschreiben die Konfiguration eines internen Sensorkomponenten, bezogen auf die Stabilisations- und Kompensationsalgorithmen.

Diese Werte können mittels Erteilung einer **POST** Anfrage entgegen `/api/sensor/detection-profiles/current/autogain` bestimmt werden. Das Ergebnis ist ein passendes Set aus Kompensationseinstellungen für diesen Sensor unter den aktuellen Umständen.

Der Inhalt dieses Datenobjekts ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).

**1.3.43 CorrectedColor**

Darstellung einer Farbe im Farbraum XYZ.

**Eigenschaften**

<p><b>values</b> Array of number, minimum items: 3, maximum items: 3, required</p>	<p>Ort in einem Farbraum</p>
--	------------------------------

### 1.3.44 Aktuelle Erkennungsprofil-ID

Der Sensor kann mehrere Erkennungsprofile speichern, aber er kann immer nur eines auf einmal anwenden. Das Feld `current_profile_id` enthält die UUID des Erkennungsprofils, das derzeit vom Sensor für seinen Betrieb verwendet wird. Es ermöglicht die Verwendung des Verknüpfungs-API-Endpunkts `/api/sensor/detection-profiles/current` anstelle der Angabe eines Erkennungsprofils anhand seiner UUID.

#### Typinformationen

[CurrentDetectionProfileID](#) (string), pattern: `^[a-f0-9-]+$`

#### Beispiele

`a014e415-0fec-4734-ac3f-30da0a5f3899`

### 1.3.45 Aktueller Schaltausgänge-Zustand

Derzeit aktiver Zustand der Schaltausgänge. Beachten Sie, dass dies von den angegebenen Ausgangszuständen des aktuell besten Matcher abweichen kann, da Einstellungen wie *ausgelöster Eingang* oder *Haltezeit* den Aktualisierungsprozess für die Schaltausgänge beeinflussen.

#### Eigenschaften

<b>states</b> Array of any of boolean or null, <b>required</b>	Liste von True/False/Null Werten beschreibt den gewollten Zustand des Schaltausgangs
---	--

### 1.3.46 Zylinder-Farbtoleranz

#### Eigenschaften

<b>limits</b> object, <b>required</b>	limits
	<b>radius</b> number, <b>required</b>
	<b>half_height</b> number, <b>required</b>
<b>shape</b> <a href="#">ToleranceShapeName</a> (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .

### 1.3.47 Standardwertepaare

#### Eigenschaften

<b>uuid</b> <a href="#">UUID</a> (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
<b>object_type</b> string, <b>required</b> , read-only	Name des Objekts das für den Standard gedacht ist
<b>key</b> string, <b>required</b> , read-only	Name der Objekteigenschaft
<b>value</b> any, <b>required</b>	Aktueller Standardwert für die Objekteigenschaft

#### Beispiele

Farbgruppe (Matcher): Toleranz

```
{
  "uuid": "a7bd36b3-e9c1-4f60-8d7e-cf47634a28b1",
  "object_type": "matcher",
  "key": "tolerance",
  "value": {
    "shape": "sphere",
    "limits": {
```

```

        "radius": 4
    }
}

```

Farbgruppe (Matcher): Haltezeit

```

{
  "uuid": "55b35901-1ea6-4b3d-864a-60af15a9b0c5",
  "object_type": "matcher",
  "key": "hold_time",
  "value": 0
}

```

Farbgruppe (Matcher): Setze Ausgabe, nach Haltezeit-Verfall zurück

```

{
  "uuid": "9ba8a7a4-7fa5-4bfc-8883-98d7b6084e91",
  "object_type": "matcher",
  "key": "reset_output_after_hold_time_expired",
  "value": false
}

```

Autogain: Anzahl an Stichproben, welche zur Mittelwertbildung genutzt wurden

```

{
  "uuid": "eeb46031-10e5-4f13-901a-c7eb16aa0cf9",
  "object_type": "autogain",
  "key": "averages",
  "value": 0
}

```

### 1.3.48 Erkennungsprofil

Ein Erkennungsprofil erkennt ein komplettes Set aus Sensoreinstellungen für eine gegebene Erkennungsaufgabe.

Mehrere Profile können gespeichert werden, um zwischen verschiedenen Erkennungsaufgaben zu wechseln oder für die inkrementelle Entwicklung eines verfeinerten Profils.

Manche Attribute eines Erkennungsprofils legen interne Details des Sensors offen, welche indirekt durch andere Mittel bestimmt werden sollten. Diese Attribute werden nur oberflächlich beschrieben, da sie so behandelt werden sollen wie sie sind ohne ihre Werte oder Struktur zu ändern.

#### Eigenschaften

<b>uuid</b> UUID (string), pattern: ^[a-f0-9-]+\$, <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8		
<b>alias</b> Alias (integer), <b>required</b> , read-only	Ein numerischer Wert, welcher benutzt werden kann um einen Gegenstand in einer Sammlung anzusprechen. Wenn ein alias neben einem uuid Attribut angegeben wird, kann dieses alias als Alternative genutzt werden um die Gegenstände in URLs und anderen Protokollen wie Modbus oder Schnittstellen anzusprechen.		
<b>name</b> String, <b>required</b>	Visuell lesbarer Name des Erkennungsprofils		
<b>colorspace</b> Colorspace, <b>required</b>	Ein Farbraum beschreibt die numerische Umwandlung von Far-		

	ben unter bestimmten Bedingungen. Verschiedene standardisierte Farbräume sind passend für verschiedene Erkennungsaufgaben.		
	colorspace		
	<b>name</b> String, <b>required</b>		
	<b>space_id</b> ColorspaceID, <b>required</b>	Eindeutiger Name eines Farbraums	
	<b>axes</b> Array of ColorspaceAxis, minimum items: 3, maximum items: 3, <b>required</b>	ColorspaceAxis[]	
		<b>id</b> String, <b>required</b>	Eindeutiger Name
		<b>label</b> String, <b>required</b>	Visuell lesbarer Name
		<b>minimum</b> Number, <b>required</b>	Niedrigster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
		<b>maximum</b> Number, <b>required</b>	Höchster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
<b>non_matching_output</b> WantedSwitchingOutputsState, <b>required</b>	Dieser Zustand der Schaltausgänge wird angewandt, wenn die aktuelle Stichprobenfarbe zu keiner gespeicherten Farbgruppe (Matcher) gehört.		
	non_matching_output		
	<b>uuid</b> UUID (string), pattern: ^[a-f0-9-]+\$, <b>required</b> , read-only	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8	
	<b>states</b> Array of any of boolean or null, <b>required</b>	Liste von True/False/Null Werten beschreibt den gewollten Zustand des Schaltausgangs	
<b>non_matching_hold_time</b> HoldTime (number), maximum 3153600000, <b>required</b>	Minimale Dauer (in Sekunden) der Anwendung des <i>non_matching_output</i> Zustands am Schaltausgang des Sensors. Diese Verlängerung eines potentiellen <i>nicht passenden</i> Falls könnte nützlich sein, wenn die Bearbeitungszeit eines verbundenen Aktors den Stichprobenzeitraum des Sensors überschreitet.		
<b>compensation_settings</b> CompensationSettings, <b>required</b>	Die Kompensationseinstellungen eines Erkennungsprofils beschreiben die Konfiguration eines internen Sensorkomponenten, bezogen auf die Stabilisations- und Kompensationsalgorithmen. Diese Werte können mittels Erteilung einer <b>POST</b> Anfrage entgegen <code>/api/sensor/detection-profiles/current/autogain</code> bestimmt werden. Das Ergebnis ist ein passendes Set aus Kompensationseinstellungen für diesen Sensor unter den aktuellen Umständen.		

	Der Inhalt dieses Datenobjekts ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).		
	<b>compensation_settings</b>		
<b>sampling_settings</b> SamplingSettings, required	Stichprobeneinstellungen beschreiben alle Details der Stichprobenentnahme. Ihre Attribute können abgefragt und untersucht werden (z.B. um die aktuelle Abtastrate abzurufen). Die meisten Werte die in den Stichprobeneinstellungen gespeichert sind sollten nicht direkt geändert werden. Der verbundene API Endpunkt <code>/api/sensor/detection-profiles/current/autogain</code> sollte stattdessen benutzt werden. Das einzige veränderbare Attribut innerhalb der Stichprobeneinstellungen ist der Durchschnittswert. Es ist sicher ihn zu ändern, obwohl die berechneten Standardwerte während eines Autogain-Betriebs optimal für die meisten Erkennungsaufgaben sein sollten.		
	<b>sampling_settings</b>		
	<b>led_intensity</b> Number, minimum: 0, maximum: 1, required	Relative Intensität des Emitters während der Leuchtphase	
	<b>base_sample_rate</b> SampleRate (number), minimum: 0.01, required	Die Basisabtastrate bestimmt die Dauer der Abtastperiode. Nach jeder Abtastperiode, werden die erfassten Daten verarbeitet und ein neues Erfassungsergebnis wird berechnet (z.B. die passendste Farbgruppe (Matcher) für das gegebene Beispiel). Dies könnte den Zustand der Schaltausgänge beeinträchtigen oder konfigurierte Aktionen auslösen. Somit definiert die Basisabtastrate die Maximumrate der Änderungen für die Schaltausgänge. Siehe ebenso die effektive Abtastrate.	
	<b>effective_sample_rate</b> SampleRate (number), minimum: 0.01, required	Die effektive Abtastrate ist ein numerisches Produkt der Basisabtastrate und der Anzahl der Durchschnitte. Sie bestimmt die Minimumdauer die ein Ziel benötigt um abgetastet zu werden um dessen visuellen Auftritt korrekt zu bestimmen. Mit dem Einstellen der Standardwerte auf eins, ist dieser Wert gleich zu der Basisabtastrate.	
	<b>minimum_wanted_sample_rate</b> SampleRate (number), minimum: 0.01, required	Dieser informative Wert repräsentiert die Abtastrate, welche während des aktuellsten Autogain-Vorgangs angefragt wurde. Die effektive Abtastrate könnte von der gewollten Abtastrate abweichen, falls die angefragte Abtastrate aufgrund der Grenzen des Sensors nicht erreichbar war (z.B. Überschreiten der unterstützten Abtastrate) oder aufgrund der Umwelt (z.B. nicht genug Licht, sodass eine langsamere Amplifikation mit höherer Verstärkung nötig war).	
	<b>sample_light_phase</b> Boolean, required	Definiert ob der Sensor die internen Emitters zum Abtasten periodisch aktivierten soll	

	<b>sample_dark_phase</b> Boolean, <b>required</b>	Definiert ob der Sensor die internen Emitter zum Abtasten periodisch deaktivieren soll.	
	<b>averages</b> AverageSampleCount (integer), minimum: 1, <b>required</b>	Anzahl vorheriger Stichproben die für jedes Stichprobenergebnis gemittelt werden sollen. Ein rollierender Mittelungsalgorithmus wird an den Stichproben angewandt.	
	<b>amplification</b> AmplificationLevel (integer), <b>required</b>	Der Amplifikationslevel gibt die interne Konfiguration eines Verstärkers an. Der Wert ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).	
<b>white_reference</b> Array of number, <b>required</b>	Das Weißreferenzattribut wird benutzt um einen benutzerdefinierten Farbausgleich anzugeben. Sein Inhalt unterliegt dem internen Gebrauch. Somit sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte (z.B. /api/sensor/detection-profiles/{itemId}/white-reference).		
<b>normalization_constant</b> Array of number, <b>required</b>	Normierungskonstanten sind verbunden mit der Weißreferenz. Ihr Inhalt unterliegt dem internen Gebrauch. Somit sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte (z.B. /api/sensor/detection-profiles/{itemId}/white-reference).		

## Beispiele

```
{
  "name": "#0",
  "uuid": "2475df8d-85f0-4208-ba60-dce6cb282a96",
  "alias": 1,
  "non_matching_hold_time": 0,
  "colorspace": {
    "name": "L*a*b*",
    "axes": [
      {
        "id": "L",
        "label": "L*",
        "minimum": 0,
        "maximum": 100
      },
      {
        "id": "a",
        "label": "a*",
        "minimum": -500,
        "maximum": 500
      },
      {
        "id": "b",
        "label": "b*",
        "minimum": -200,
        "maximum": 200
      }
    ]
  },
  "space_id": "Lab"
}
```

```

},
"compensation_settings": {
  "monitor_integration": {
    "control": 0.32499998807907104,
    "references": [
      0.7283520102500916,
      0.7442666888237,
      0.7066696286201477
    ]
  },
  "use_calibration_samples": true
},
"normalization_constant": [
  237.4935277662995,
  242.62655153828055,
  587.8264132734112
],
"white_reference": [
  95.047,
  100,
  108.883
],
"non_matching_output": {
  "uuid": "3f26aff4-8650-42a0-b319-51776c443fbc",
  "states": [
    true,
    true,
    true,
    true,
    true,
    true,
    true
  ]
},
"sampling_settings": {
  "led_intensity": 1,
  "amplification": 1,
  "sample_light_phase": true,
  "minimum_wanted_sample_rate": 1000,
  "averages": 1,
  "base_sample_rate": 1000,
  "sample_dark_phase": true,
  "effective_sample_rate": 1000
}
}

```

### 1.3.49 Geräteinformationen

#### Eigenschaften

<b>id</b> DeviceSerialNumber, required	Serien Nummer
<b>model_name</b> string, required	lesbarer Name des Controller Modells
<b>model_key</b> string, required	eindeutige ID des Controller modells
<b>variant</b> any of string or null, required	zeigt eine spezielle Option eines Controller Modells an
<b>vendor_key</b> DeviceVendorKey, required	Eindeutiger Schlüssel, der den Hersteller identifiziert, der dieses Gerät vertreibt
<b>vendor_name</b> DeviceVendorName, required	Name des Herstellers dieses Geräts
<b>device_id</b> DeviceSerialNumber, optional, Deprecated	Veraltet: Verwenden Sie stattdessen "id" (wird in zukünftigen Versionen verschwinden)

<b>model</b> string, optional, Deprecated	Veraltet: Verwenden Sie stattdessen "model_name" (wird in zukünftigen Versionen verschwinden)
<b>vendor</b> DeviceVendorName, optional, Deprecated	Veraltet: Verwenden Sie stattdessen "vendor_name"

## Beispiele

```
{
  "vendor_name": "Micro-Epsilon Eltrotec GmbH",
  "vendor_key": "eltrotec",
  "variant": null,
  "model_key": "me_cfo_100",
  "model_name": "CFO100",
  "id": "7454228060"
}
```

### 1.3.50 Geräteseriennummer

Seriennummer

#### Typinformation

[DeviceSerialNumber](#) (string)

### 1.3.51 Geräteherstellerschlüssel

Eindeutiger Schlüssel, der die Organisation identifiziert, die dieses Gerät verteilt.

#### Typinformationen

[DeviceVendorKey](#) (string)

#### Beispiele

acme

### 1.3.52 Geräteherstellername

Name des Herstellers dieses Geräts.

#### Typinformationen

[DeviceVendorName](#) (string)

#### Beispiele

Acme Corporation

### 1.3.53 Fehler

Liste von Fehlerindikatoren, die sowohl maschinenlesbar als auch visuell lesbar sind.

#### Eigenschaften

<b>code</b> string, optional	maschinenlesbarer eindeutiger Fehlercode
<b>mapping</b> string, optional	ein Verweis auf den Parameter, der den Fehler verursacht hat
<b>message</b> string, optional	vom Menschen lesbare Fehlerbeschreibung

### 1.3.54 Firmware-Build-ID

Eindeutige ID des aktuell laufenden Firmware-Images.

colorSENSOR CFO

## Typinformationen

[FirmwareBuildId](#) (string), pattern: `^[a-f0-9-]+$`

## Beispiele

d985c28e03a4eb39132c02affeb29e71

### 1.3.55 Firmware-Image-Datei

## Typinformationen

[FirmwareImageFile](#) (file)

### 1.3.56 Firmware-Image Größe

Größe des Firmware-Images in Byte

## Typinformationen

[FirmwareImageSize](#) (integer), minimum: 1, maximum: 1073741824

### 1.3.57 Firmware-Image Upload

Ein vollständig oder teilweise hochgeladenes Firmware-Image, das für die Aktualisierung der Firmware verwendet wird

## Eigenschaften

<b>uuid</b> UUID (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b> , read-only	Eindeutige ID eines Firmware-Uploads
<b>build_id</b> HashDigest (string), pattern: <code>^[a-f0-9-]+\$</code> , <b>required</b>	Eindeutige ID des aktuell laufenden Firmware-Images
<b>status</b> string, one of [incomplete, complete, invalid_signature, processing_failure, malformed_content, device_mismatch], <b>required</b>	Aktueller Status des Firmware-Uploads  unvollständig Die Anzahl der empfangenden Bytes ist niedriger als die Anzahl der Bytes, welche vollständig angekündigt wurden. Der Firmware-Upload ist vollständig und neue Firmware kann angewandt werden invalid_signature. Die Firmware-Prüfsumme passte nicht zum erwarteten Wert processing_failure ein interner, undefinierter Fehler trat, während der Verarbeitung der Firmware auf malformed_content. Das hochgeladene Firmware-Image nutzt ein unerwartetes Format oder es fehlen essentielle Informationen device_mismatch. Das Firmware-Image kann auf diesem Gerät nicht angewandt werden,
<b>uploaded_size</b> integer, minimum: 0, <b>required</b>	Anzahl hochgeladener Bytes
<b>expected_size</b> integer, minimum: 1, <b>required</b>	erwartete Gesamtanzahl an Bytes für das Firmware-Image
<b>max_chunk_size</b> integer, minimum: 1, <b>required</b>	maximale Größe für einen hochladbaren Datensatz auf dem Gerät

## Beispiele

```
{
  "uuid": "78b40d5e-e82c-45a9-8842-9481f889f790",
  "build_id": "e943ce84dbe474bc4d163b44c90070b105fd66bb",
  "expected_size": 335544320,
  "max_chunk_size": 1048576,
  "status": "incomplete",
```

```
"uploaded_size": 24117248
}
```

### 1.3.58 Firmware-Informationen

Informationen die die Firmware-Version beschreiben.

#### Eigenschaften

<b>id</b> <i>FirmwareBuildId</i> (string), pattern: <code>^[a-f0-9]+\$</code> , <b>required</b>	Eindeutige ID des aktuell laufenden Firmware-Images
<b>channel</b> <i>ReleaseChannel</i> (string), one of [stable, feature], default: stable, <b>required</b>	Beschreibt die Art der Publikation  Veröffentlichungen auf dem <i>stable</i> Kanal, gelten generell als gut getestet und werden für die Nutzung in der Produktion empfohlen. Veröffentlichungen auf dem <i>feature</i> Kanal, fügen neue Funktionen hinzu aber wurden nicht so oft wie stable Veröffentlichungen getestet. Feature Veröffentlichungen können, aber sollten nur mit reichlicher Überlegung in der Produktion genutzt werden.
<b>created_on</b> <i>Timestamp</i> (string), <b>required</b>	Zeit in der diese Firmware erstellt wurde
<b>name</b> string, <b>required</b>	Visuell lesbarer Name dieser Veröffentlichung
<b>notes</b> string, <b>required</b>	Versionshinweise formatiert als Abschlag
<b>version</b> <i>FirmwareVersion</i> (string), <b>required</b>	Version einer Firmware
<b>works_with</b> Array of string, <b>required</b>	Kompatible Gerätemodelle (siehe <i>model_key</i> in /api/device)

### 1.3.59 Firmware-Wiederherstellungsinformationen

#### Typinformationen

[FirmwareRecoveryInformation](#) (string)

#### Beispiele

```
{
  "created_on": "2018-02-13T05:40:39+01:00",
  "name": "CFO",
  "id": "4fab356b5014b5cc82efc4a81bfefbfcdc2d9165",
  "version": "1.3.16",
  "channel": "stable",
  "works_with": [
    "me_cfo_100",
    "me_cfo_200"
  ],
  "notes": "# Release 1.3.16 (2018-02-13 - CFO)\n\n## Veröffentlichungshinweise\n\nWartungsrelease für CFO-Sensoren.\n\n\n## Änderungen\nkeine\n\n\n## Fehlerkorrekturen\n* Hochladen von Konfigurationsdateien mit mehr als 70 Farben ermöglicht\n* Announcierung des korrekten Hostnamen via avahi/zeroconf\n* SSDP: Kommunikation via IPv6-Link-Local-Adresse ermöglicht\n* SSDP: auch die Auto-Konfigurations-IP (via RFC3927) unter \"CurrentAddresses\" announcieren\n* SSDP: nach Konfigurationsänderungen an neue IP-Adressen binden"
}
```

### 1.3.60 Firmware-Ausführungsinformationen

Informationen die die aktuell ausgeführte Firmware beschreiben.

#### Eigenschaften

<b>build_id</b>	Eindeutige ID des aktuell laufenden Firmware-Images
colorSENSOR CFO	

<code>FirmwareBuildId</code> (string), pattern: <code>^[a-f0-9]+\$</code> , <b>required</b>	
<b>source_url</b> any of string or null, optional	Absolute Basis-URL eines Firmware-Repositorys, das für dieses Gerät geeignete Firmware-Abbilder liefert
<b>version</b> <code>FirmwareVersion</code> (string), <b>required</b>	Version einer Firmware

### Beispiele

```
{
  "build_id": "4fab356b5014b5cc82efc4a81bfefbfcdc2d9165",
  "source_url": null,
  "version": "1.3.16"
}
```

### 1.3.61 Firmware-Einstellungen

Einstellungen bezüglich der Geräte Firmware und Updates.

#### Eigenschaften

<b>channel</b> <code>ReleaseChannel</code> (string), one of [stable, feature], default: stable, <b>required</b>	Beschreibt die Art der Publikation Veröffentlichungen auf dem stable Kanal, gelten generell als gut getestet und werden für die Nutzung in der Produktion empfohlen. Veröffentlichungen auf dem feature Kanal, fügen neue Funktionen hinzu aber wurden nicht so oft wie stable Veröffentlichungen getestet. Feature Veröffentlichungen können, aber sollten nur mit reichlicher Überlegung in der Produktion genutzt werden.
--	--

### Beispiele

```
{
  "release_channel": "stable"
}
```

### 1.3.62 Firmware Version

Version einer Firmware.

#### Typinformationen

`FirmwareVersion` (string)

#### Beispiele

v2.3.42

### 1.3.63 Hash Digest

Eindeutiger Bezeichner (hexadezimaler digest string).

#### Typinformationen

`HashDigest` (string), pattern: `^[a-f0-9]+$`

#### Beispiele

d985c28e03a4eb39132c02affeb29e71

### 1.3.64 Haltezeit

Minstdauer (in Sekunden), während der die Ausgabeeinstellungen einer Farbgruppe (Matcher) nach der Erkennung angewendet werden.

#### Typinformationen

[HoldTime](#) (number), minimum: 0, maximum: 3153600000

### 1.3.65 Hostname

Visuell lesbarer Name, der das Gerät im Netzwerk identifiziert.

#### Typinformationen

[Hostname](#) (string), pattern: `^(?:[a-zA-Z0-9](?:[a-zA-Z0-9\-\_]*[a-zA-Z0-9])?\.)*[a-zA-Z0-9](?:[a-zA-Z0-9\-\_]*[a-zA-Z0-9])?$`

### 1.3.66 Unbegrenzte Farbtoleranz

#### Eigenschaften

<b>limits</b> Object, <b>required</b>	limits
<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .

### 1.3.67 Zustand der Eingänge

Der Zustand aller Eingänge während eines bestimmten Zeitraums wird durch eine Liste möglicher Ereignisse in Kombination mit einem booleschen Wert angegeben, der angibt, ob das gegebene Ereignis innerhalb des Zeitraums aufgetreten ist.

#### Eigenschaften

// boolean, <b>required</b>	Der boolesche Wert gibt an, ob das benannte Eingabeereignis in der letzten Periode aufgetreten ist.
--------------------------------	---

### 1.3.68 Schnittstelle RS232

#### Eigenschaften

<b>protocol</b> any of <a href="#">SerialModbusSettings</a> or <a href="#">SerialElizaSettings</a> , <b>required</b>	SerialModbusSettings
	<b>type</b> string, one of [none, eliza, modbus] , default: eliza, <b>required</b>
	<b>slave_id</b> any of number or null, <b>required</b>
	<b>frame_format</b> string, one of [rtu, ascii] , default: rtu, <b>required</b>
	SerialElizaSettings
	<b>type</b> string, one of [none, eliza, modbus] , default: eliza, <b>required</b>
<b>baud_rate</b> number, one of [9600, 19200, 115200], <b>required</b>	

### 1.3.69 Schnittstelle USB

#### Eigenschaften

<b>protocol</b> any of <a href="#">SerialModbusSettings</a> or <a href="#">SerialElizaSettings</a> , <b>required</b>	SerialModbusSettings
	<b>type</b> string, one of [none, eliza, modbus] , default: eliza, <b>required</b>
	<b>slave_id</b> any of number or null, <b>required</b>
	<b>frame_format</b> string, one of [rtu, ascii] , default: rtu, <b>required</b>
	SerialElizaSettings

	<b>type</b> string, one of [none, eliza, modbus] , default: eliza, <b>required</b>
<b>baud_rate</b> number, one of [9600, 19200, 115200], <b>required</b>	

### 1.3.70 Tastatur-Ereignis

Ein Tastatur-Ereignis stellt ein einzelnes Drücken oder Loslassen einer Taste zu einer bestimmten Zeit dar.

#### Eigenschaften

<b>source</b> string, <b>required</b>	Die übliche Quelle der Ereignisse ist <i>inputs</i> .
<b>name</b> <a href="#">KeypadEventInput</a> (string), <b>required</b>	Name einer Tastatureingabe (Knopf), welcher Ereignisse auslösen könnte.
<b>event</b> <a href="#">KeypadEventName</a> (string), <b>required</b>	Eingangspерipherie kann verschiedene Ereignisse auslösen.
<b>timestamp</b> integer, minimum: 0, <b>required</b>	Der Zeitstempel ist in Millisekunden gegeben und sollte monoton steigen.

#### Beispiele

Intensität Knopf gedrückt:

```
{
  "source": "inputs",
  "name": "intensity",
  "event": "down",
  "timestamp": 6403500
}
```

Intensität Knopf losgelassen:

```
{
  "source": "inputs",
  "name": "intensity",
  "event": "up",
  "timestamp": 6405800
}
```

### 1.3.71 Tastatur-Ereignis-Eingabe

Name einer Tastatureingabe (Taste), die Ereignisse auslösen kann.

#### Typinformationen

[KeypadEventInput](#) (string)

### 1.3.72 Tastatur-Ereignis-Name

Eingangspерipheriegeräte können verschiedene Ereignisse auslösen.

#### Typinformationen

[KeypadEventName](#) (string)

### 1.3.73 Tastaturanzeige

Die Tastatur verfügt über mehrere LEDs als visuelle Anzeigen.

Die Anzeigen können leuchten, blinken oder aus sein.

#### Eigenschaften

colorSENSOR CFO

<b>name</b> string, required	Name der Anzeige	
<b>type</b> string, required	Der Typ beschreibt die möglichen Visualisierungsmodi für diese Anzeige.	
<b>animation</b> Array of object, required	Der visuelle Zustand jeder Anzeige wird durch eine Endlosschleife von Animationsschritten beschrieben.	
	object []	
	<b>enabled</b> boolean, required	Visueller Status der Anzeige (an oder aus)
	<b>color</b> string, optional	Name oder Beschreibung einer Farbe
	<b>duration</b> number, required	Dauer (in Sekunden) dieses Teils der Schleifenanimation

### Beispiele

```
{
  "name": "trigger_teach",
  "type": "colored",
  "animation": [
    {
      "enabled": true,
      "color": "green",
      "duration": 0.6
    },
    {
      "enabled": false,
      "duration": 0.4
    }
  ]
}
```

### 1.3.74 Tastaturinformationen

Beschreibt den aktuellen Zustand der Tastatur sowieso den Zugriff auf die Visualisierungsdaten.

#### Eigenschaften

<b>locked</b> boolean, required	Boolean-Kennzeichen, welches Status der Tastatursperre angibt (true -> gesperrt, false -> entsperrt). Alle Tastatureingaben werden ignoriert solange die Sperre aktiv ist.
<b>clear_matcher_before_teach</b> boolean, required	Das Boolean-Kennzeichen steuert ob mehrere Farben für eine Farbgruppe durch Tastatur-basierte Lernvorgänge gespeichert werden können. Ein Wert aus true impliziert, dass ein Lernvorgang alle existierenden Farben aus der momentan ausgewählten Farbgruppe entfernt, bevor neue Farben hinzugefügt werden. Mit einem Wert von false, werden vorher existierende Farben nicht gelöscht bevor eine neue hinzugefügt wird.
<b>visualization_url</b> any of string or null, optional, read-only	Der Visualisierungsquellenstandort, kann für das Bereitstellen einer virtuellen Tastaturschnittstelle genutzt werden. Seine URL kann mit einem Schema (z.B. http oder https) für eine volle URL mit Hostnamen starten oder sie startet mit einem Schrägstrich, hinweisend auf einen Weg, welcher von dem Gerät selbst zur Verfügung gestellt wird. Dieses Attribut kann nicht geändert werden.

### Beispiele

```
{
  "locked": true,
  "clear_matcher_before_teach": false,
  "visualization_url": "/media/keypad-image.svg"
}
```

### 1.3.75 Tastatureingabetaste

Die Tastatur enthält mehrere Eingänge (Tasten), die Ereignisse erzeugen können.

#### Eigenschaften

<b>name</b> KeypadEventInput (string), <b>required</b>	Name einer Tastatureingabe (Knopf), welcher Ereignisse auslösen könnte.	
<b>capabilities</b> Array of object, <b>required</b>	object[]	
	<b>name</b> KeypadEventName (string), <b>required</b>	Eingangssperipherie kann verschiedene Ereignisse auslösen.
	<b>url</b> string, <b>required</b>	Die Ereignisse können extern durch Abgabe einer <b>POST</b> -Anfrage entgegen dieser Quelle ausgelöst werden.

#### Beispiele

```
{
  "name": "intensity",
  "capabilities": [
    {
      "name": "down",
      "url": "/api/peripherals/keypad/inputs/intensity/down"
    },
    {
      "name": "up",
      "url": "/api/peripherals/keypad/inputs/intensity/up"
    }
  ]
}
```

### 1.3.76 Anmeldeinformationen

Beschreibt die derzeit aktive Anmeldung durch den User-Agent.

#### Eigenschaften

<b>logged_in_user</b> any of <a href="#">User</a> or null, <b>required</b>	Der aktuell angemeldete Benutzer. Ist <code>null</code> , wenn die Anmeldedaten zu keinem bekannten Benutzer passen oder abgelaufen sind.	
	<b>name</b> string, pattern: <code>^[w-]+\$</code> , <b>required</b> , <b>read-only</b>	Eindeutiger Name, der ein Konto identifiziert
	<b>password</b> string, optional	Diesem Konto zugewiesenes Passwort (nur beschreibbar; wird in Antworten nie zurückgegeben). Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein <code>password</code> oder ein <code>password_hash</code> angegeben werden.
	<b>password_hash</b> <a href="#">HashDigest</a> (string), pattern: <code>^[a-f0-9]+\$</code> , optional	Diesem Konto zugewiesener Passwort-Hash. Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein <code>password</code> oder ein <code>password_hash</code> angegeben werden.
	<b>roles</b> Array of string, optional	Die einem Konto zugewiesenen Rollen, definieren den Berechtigungssatz des Kontos
<b>session_timeout</b> any of number or null, <b>required</b>	Anzahl der Sekunden, die dieser Sitzung noch verbleiben, bevor sie abläuft. Ist <code>null</code> , wenn die angegebenen Anmeldedaten keiner aktiven Sitzung zugeordnet werden konnten, wenn die Sitzung abgelaufen ist oder wenn der angegebene Authentifizierungsmechanismus keine Sitzungen unterstützt (z.B. HTTP-Authentifizierung).	

### 1.3.77 MAC-Adresse

Eindeutige Hardware-Adresse einer Netzwerkschnittstelle

#### Typinformationen

[MacAddress](#) (string), pattern: `^([a-f0-9]{2}:){5}[a-f0-9]{2}$`

#### Beispiele

00:01:2e:7a:dc:23

### 1.3.78 Netzwerk-Adresskonfiguration IPv4

#### Eigenschaften

<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
---	---

### 1.3.79 Netzwerk-Adresskonfiguration IPv4DHCP

Das Dynamic Host Configuration Protocol erfordert einen Router, der Leases auf Anfrage verteilt.

#### Eigenschaften

<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
---	---

### 1.3.80 Netzwerk-Adresskonfiguration IPv4Static

Die statische Adressenkonfiguration ist nicht von der Netzwerkinfrastruktur abhängig.

#### Eigenschaften

<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
<b>address</b> <a href="#">NetworkInterfaceAddressIPv4</a> (string), <b>required</b>	IPv4 Netzwerkadresse in CIDR-Notation
<b>gateway</b> <a href="#">NetworkAddressIPv4</a> (string), optional	Standard Gateway für abgehenden Verkehr

### 1.3.81 Netzwerk-Adresskonfiguration IPv6

#### Eigenschaften

<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
---	---

### 1.3.82 Netzwerk-Adresskonfiguration IPv6Auto

Die zustandslose Adressen-Autokonfiguration (SLAAC) beruht ausschließlich auf dem Neighbourhood Discovery Protocol. SLAAC ist nur für IPv6 verfügbar.

#### Eigenschaften

<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
---	---

### 1.3.83 Netzwerk-Adresskonfiguration IPv6DHCP

Das Dynamic Host Configuration Protocol erfordert einen Router, der Leases auf Anfrage verteilt.

#### Eigenschaften

<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
---	---

### 1.3.84 Netzwerk-Adresskonfiguration IPv6Static

Die statische Adressenkonfiguration ist nicht von der Netzwerkinfrastruktur abhängig.

#### Eigenschaften

<b>method</b> string, one of [static, dhcp], required	Konfigurationsmethode, welche für die Adresse benutzt wird.
<b>address</b> <a href="#">NetworkInterfaceAddressIPv6</a> (string), required	IPv6 Netzwerkadresse in CIDR-Notation
<b>gateway</b> <a href="#">NetworkAddressIPv4</a> (string), optional	Standard Gateway für abgehenden Verkehr

### 1.3.85 Netzwerkadresse IPv4

#### Typinformationen

[NetworkAddressIPv4](#) (string)

#### Beispiele

192.168.1.100

### 1.3.86 Netzwerkadresse IPv6

#### Typinformationen

[NetworkAddressIPv6](#) (string)

#### Beispiele

fd00::576b:c643:100:40f:10ff:fe02:e6f

### 1.3.87 Informationen zur Konfiguration der Netzwerkschnittstellen-Adresse

#### Eigenschaften

<b>ipv4</b> <a href="#">NetworkInterfaceAddressFamilyInformationIPv4</a> , required	IPv4 Netzwerkadresskonfiguration		
	<b>ipv4</b>		
	<b>address_configurations</b> Array of any of <a href="#">NetworkAddressConfigurationIPv4Static</a> or <a href="#">NetworkAddressConfigurationIPv4DHCP</a> , optional	<a href="#">NetworkAddressConfigurationIPv4Static</a> []	
		<b>method</b> string, one of [static, dhcp], required	Konfigurationsmethode, welche für die Adresse benutzt wird.
		<b>address</b> <a href="#">NetworkInterfaceAddressIPv4</a> (string), required	IPv4 Netzwerkadresse in CIDR-Notation
		<b>gateway</b> <a href="#">NetworkAddressIPv4</a> (string), optional	Standard Gateway für abgehenden Verkehr
		<a href="#">NetworkAddressConfigurationIPv4DHCP</a> []	
		<b>method</b> string, one of [static, dhcp], required	Konfigurationsmethode, welche für die Adresse benutzt wird.
	<b>current_addresses</b> Array of <a href="#">WrappedNetworkInterfaceAddressIPv4</a> , required	<a href="#">WrappedNetworkInterfaceAddressIPv4</a> []	

		<b>address</b> <a href="#">NetworkInterfaceAddressIPv4</a> (string), <b>required</b>	IPv4 Netzwerkadresse in CIDR Notation
<b>ipv6</b> <a href="#">NetworkInterfaceAddressFamilyInformationIPv6</a> , <b>required</b>	IPv6 Netzwerkadresskonfiguration		
	ipv6		
	<b>address_configurations</b> Array of any of <a href="#">NetworkAddressConfigurationIPv6Static</a> , <a href="#">NetworkAddressConfigurationIPv6DHCP</a> or <a href="#">NetworkAddressConfigurationIPv6Auto</a> , optional	<a href="#">NetworkAddressConfigurationIPv6Static</a> []	
		<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
		<b>address</b> <a href="#">NetworkInterfaceAddressIPv6</a> (string), <b>required</b>	IPv4 Netzwerkadresse in CIDR-Notation
		<b>gateway</b> <a href="#">NetworkAddressIPv6</a> (string), optional	Standard Gateway für abgehenden Verkehr
		<a href="#">NetworkAddressConfigurationIPv6DHCP</a> []	
		<b>method</b> string, one of [static, dhcp, auto], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
		<a href="#">NetworkAddressConfigurationIPv6Auto</a> []	
		<b>method</b> string, one of [static, dhcp, auto], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
	<b>current_addresses</b> Array of <a href="#">WrappedNetworkInterfaceAddressIPv6</a> , <b>required</b>	<a href="#">WrappedNetworkInterfaceAddressIPv6</a> []	
		<b>address</b> <a href="#">NetworkInterfaceAddressIPv6</a> (string), <b>required</b>	IPv6 Netzwerkadresse in CIDR Notation

### 1.3.88 Konfiguration der Netzwerkschnittstellen-Adresse IPv4

#### Typinformationen

Array of any of [NetworkAddressConfigurationIPv4Static](#) or [NetworkAddressConfigurationIPv4DHCP](#)

### 1.3.89 Konfiguration der Netzwerkschnittstellen-Adresse IPv6

#### Typinformationen

Array of any of [NetworkAddressConfigurationIPv6Static](#), [NetworkAddressConfigurationIPv6DHCP](#) or [NetworkAddressConfigurationIPv6Auto](#)

### 1.3.90 Konfigurationszustand der Netzwerkschnittstellenadresse

Konfigurierbare Netzwerk-Adresskonfiguration einer Netzwerkschnittstelle

#### Eigenschaften

<b>ipv4</b> <u>NetworkInterfaceAddressFamilyStateIPv4</u> , optional	IPv4 Netzwerkadresskonfiguration		
	ipv4		
	<b>address_configurations</b> Array of any of <u>NetworkAddressConfigurationIPv4Static</u> or <u>NetworkAddressConfigurationIPv4DHCP</u> , optional	NetworkAddressConfigurationIPv4Static[]	
		<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
		<b>address</b> <u>NetworkInterfaceAddressIPv4</u> (string), <b>required</b>	IPv4 Netzwerkadresse in CIDR-Notation
		<b>gateway</b> <u>NetworkAddressIPv4</u> (string), optional	Standard Gateway für abgehenden Verkehr
		NetworkAddressConfigurationIPv4DHCP[]	
		<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
<b>ipv6</b> <u>NetworkInterfaceAddressFamilyStateIPv6</u> , optional	IPv6 Netzwerkadresskonfiguration		
	ipv6		
	<b>address_configurations</b> Array of any of <u>NetworkAddressConfigurationIPv6Static</u> , <u>NetworkAddressConfigurationIPv6DHCP</u> or <u>NetworkAddressConfigurationIPv6Auto</u> , optional	NetworkAddressConfigurationIPv6Static[]	
		<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
		<b>address</b> <u>NetworkInterfaceAddressIPv6</u> (string), <b>required</b>	IPv6 Netzwerkadresse in CIDR-Notation
		<b>gateway</b> <u>NetworkAddressIPv6</u> (string), optional	Standard Gateway für abgehenden Verkehr
		NetworkAddressConfigurationIPv6DHCP[]	
		<b>method</b> string, one of [static, dhcp, auto], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
		NetworkAddressConfigurationIPv6Auto[]	
		<b>method</b> string, one of [static, dhcp, auto], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.

**Beispiele**

Entferne alle IPv6 Adresskonfigurationen

```
{
  "ipv6": {
    "address_configurations": []
  }
}
```

Ersetze alle existierenden IPv4 Konfigurationen mit DHCP

```
{
  "ipv4": {
    "address_configurations": [
      {
        "method": "dhcp"
      }
    ]
  }
}
```

Setze statische und dynamische IPv4 Konfiguration

```
{
  "ipv4": {
    "address_configurations": [
      {
        "method": "dhcp"
      },
      {
        "method": "static",
        "address": "192.168.0.100/24"
      }
    ]
  }
}
```

### 1.3.91 Netzwerkschnittstellenadressen IPv4

#### Eigenschaften

<b>current_addresses</b> Array of <u>WrappedNetworkInterfaceAddressIPv4</u> , <b>required</b>	WrappedNetworkInterfaceAddressIPv4[]	
	<b>address</b> <u>NetworkInterfaceAddressIPv4</u> (string), <b>required</b>	IPv4 Netzwerkadresse in CIDR Notation

### 1.3.92 Netzwerkschnittstellenadressen IPv6

#### Eigenschaften

<b>current_addresses</b> Array of <u>WrappedNetworkInterfaceAddressIPv6</u> , <b>required</b>	WrappedNetworkInterfaceAddressIPv6[]	
	<b>address</b> <u>NetworkInterfaceAddressIPv6</u> (string), <b>required</b>	IPv6 Netzwerkadresse in CIDR Notation

### 1.3.93 Netzwerkschnittstellenadressen-Familieninformationen IPv4

Konfiguration der IPv4-Netzwerkadresse.

#### Eigenschaften

<b>address_configurations</b> Array of any of <u>NetworkAddressConfigurationIPv4Static</u> or <u>NetworkAddressConfigurationIPv4DHCP</u> , optional	NetworkAddressConfigurationIPv4Static[]	
--	---	--

	<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
	<b>address</b> <a href="#">NetworkInterfaceAddressIPv4</a> (string), <b>required</b>	IPv4 Netzwerkadresse in CIDR-Notation
	<b>gateway</b> <a href="#">NetworkAddressIPv4</a> (string), optional	Standard Gateway für abgehenden Verkehr
	<a href="#">NetworkAddressConfigurationIPv4DHCP</a> []	
	<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
<b>current_addresses</b> Array of <a href="#">WrappedNetworkInterfaceAddressIPv4</a> , <b>required</b>	<a href="#">WrappedNetworkInterfaceAddressIPv4</a> []	
	<b>address</b> <a href="#">NetworkInterfaceAddressIPv4</a> (string), <b>required</b>	IPv4 Netzwerkadresse in CIDR Notation

### Beispiele

```
{
  "address_configurations": [
    {
      "method": "static",
      "address": "169.254.168.150/16"
    }
  ],
  "current_addresses": [
    {
      "address": "169.254.168.150/16"
    }
  ]
}
```

### 1.3.94 Netzwerkschnittstellenadressen-Familieninformationen IPv6

Konfiguration der IPv6-Netzwerkadresse.

#### Eigenschaften

<b>address_configurations</b> Array of any of <a href="#">NetworkAddressConfigurationIPv6Static</a> , <a href="#">NetworkAddressConfigurationIPv6DHCP</a> or <a href="#">NetworkAddressConfigurationIPv6Auto</a> , optional	<a href="#">NetworkAddressConfigurationIPv6Static</a> []	
	<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
	<b>address</b> <a href="#">NetworkInterfaceAddressIPv6</a> (string), <b>required</b>	IPv6 Netzwerkadresse in CIDR-Notation
	<b>gateway</b> <a href="#">NetworkAddressIPv6</a> (string), optional	Standard Gateway für abgehenden Verkehr
	<a href="#">NetworkAddressConfigurationIPv6DHCP</a> []	
	<b>method</b> string, one of [static, dhcp, auto], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
	<a href="#">NetworkAddressConfigurationIPv6Auto</a> []	
	<b>method</b> string, one of [static, dhcp, auto], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
<b>current_addresses</b>	<a href="#">WrappedNetworkInterfaceAddressIPv6</a> []	

Array of <u>WrappedNetworkInterfaceAddressIPv6</u> , <b>required</b>		
	<b>address</b> <u>NetworkInterfaceAddressIPv6</u> (string), <b>required</b>	IPv6 Netzwerkadresse in CIDR Notation

**Beispiele**

```
{
  "address_configurations": [
    {
      "method": "static",
      "address": "fda0:576b:c643:100::100/64"
    },
    {
      "method": "auto"
    }
  ],
  "current_addresses": [
    {
      "address": "fda0:576b:c643:100::100/64"
    },
    {
      "address": "fd01::40f:10ff:fe02:e6f/64"
    },
    {
      "address": "fe80::40f:10ff:fe02:e6f/64"
    }
  ]
}
```

**1.3.95 Netzwerkschnittstellenadresse Familienzustand IPv4**

Konfiguration der IPv4-Netzwerkadresse.

**Eigenschaften**

<b>address_configurations</b> Array of any of <u>NetworkAddressConfigurationIPv4Static</u> or <u>NetworkAddressConfigurationIPv4DHCP</u> , optional	NetworkAddressConfigurationIPv4Static[]	
	<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
	<b>address</b> <u>NetworkInterfaceAddressIPv4</u> (string), <b>required</b>	IPv4 Netzwerkadresse in CIDR-Notation
	<b>gateway</b> <u>NetworkAddressIPv4</u> (string), optional	Standard Gateway für abgehenden Verkehr
	NetworkAddressConfigurationIPv4DHCP[]	
	<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.

**1.3.96 Netzwerkschnittstellenadresse Familienzustand IPv6**

Konfiguration der IPv6-Netzwerkadresse.

**Eigenschaften**

<b>address_configurations</b> Array of any of <u>NetworkAddressConfigurationIPv6Static</u> , <u>NetworkAddressConfigurationIPv6DHCP</u> or <u>NetworkAddressConfigurationIPv6Auto</u> , optional	NetworkAddressConfigurationIPv6Static[]	
---	---	--

	<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
	<b>address</b> <a href="#">NetworkInterfaceAddressIPv6</a> (string), <b>required</b>	IPv6 Netzwerkadresse in CIDR-Notation
	<b>gateway</b> <a href="#">NetworkAddressIPv6</a> (string), optional	Standard Gateway für abgehenden Verkehr
	<a href="#">NetworkAddressConfigurationIPv6DHCP</a> []	
	<b>method</b> string, one of [static, dhcp, auto], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
	<a href="#">NetworkAddressConfigurationIPv6Auto</a> []	
	<b>method</b> string, one of [static, dhcp, auto], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.

### 1.3.97 Netzwerkschnittstellenadresse IPv4

IPv4-Netzwerkadresse in CIDR-Notation.

#### Typinformationen

[NetworkInterfaceAddressIPv4](#) (string)

#### Beispiele

192.168.1.100/24

### 1.3.98 Netzwerkschnittstellenadresse IPv6

IPv6-Netzwerkadresse in CIDR-Notation.

#### Typinformationen

[NetworkInterfaceAddressIPv6](#) (string)

#### Beispiele

fd00::576b:c643:100:40f:10ff:fe02:e6f/64

### 1.3.99 Netzwerkschnittstelleninformationen

Beschreibung der aktuell aktiven Adressen der Schnittstelle und ihrer Konfiguration.

#### Eigenschaften

<b>iface</b> <a href="#">NetworkInterfaceName</a> (string), pattern: ^ [a-z0-9-]+\$, <b>required</b> , read-only	Eindeutiger Name welcher Netzwerkschnittstelle beschreibt		
<b>hardware_address</b> <a href="#">MacAddress</a> (string), pattern: ^ ([a-f0-9]{2}-){5}[a-f0-9]{2}\$, <b>required</b> , read-only	Eindeutige Hardwareadresse einer Netzwerkschnittstelle		
<b>has_link</b> boolean, <b>required</b> , read-only	Aktueller physikalischer Verbindungsstatus (ob ein Kabel eingesteckt ist oder nicht)		
<b>ipv4</b> <a href="#">NetworkInterfaceAddressFamilyStateIPv4</a> , optional	IPv4 Netzwerkadressskonfiguration		
	ipv4		
	<b>address_configurations</b> Array of any of <a href="#">NetworkAddressConfigurationIPv4Static</a>	<a href="#">NetworkAddressConfigurationIPv4Static</a> []	

	or <a href="#">NetworkAddressConfigurationIPv4DHCP</a> , optional		
		<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
		<b>address</b> <a href="#">NetworkInterfaceAddressIPv4</a> (string), <b>required</b>	IPv4 Netzwerkadresse in CIDR-Notation
		<b>gateway</b> <a href="#">NetworkAddressIPv4</a> (string), optional	Standard Gateway für abgehenden Verkehr
		<a href="#">NetworkAddressConfigurationIPv4DHCP</a> []	
		<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
<b>ipv6</b> <a href="#">NetworkInterfaceAddressFamilyStateIPv6</a> , optional	IPv6 Netzwerkadresskonfiguration		
	ipv6		
	<b>address_configurations</b> Array of any of <a href="#">NetworkAddressConfigurationIPv6Static</a> , <a href="#">NetworkAddressConfigurationIPv6DHCP</a> or <a href="#">NetworkAddressConfigurationIPv6Auto</a> , optional	<a href="#">NetworkAddressConfigurationIPv6Static</a> []	
		<b>method</b> string, one of [static, dhcp], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
		<b>address</b> <a href="#">NetworkInterfaceAddressIPv6</a> (string), <b>required</b>	IPv6 Netzwerkadresse in CIDR-Notation
		<b>gateway</b> <a href="#">NetworkAddressIPv6</a> (string), optional	Standard Gateway für abgehenden Verkehr
		<a href="#">NetworkAddressConfigurationIPv6DHCP</a> []	
		<b>method</b> string, one of [static, dhcp, auto], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.
		<a href="#">NetworkAddressConfigurationIPv6Auto</a> []	
		<b>method</b> string, one of [static, dhcp, auto], <b>required</b>	Konfigurationsmethode, welche für die Adresse benutzt wird.

### Beispiele

Entferne alle IPv6 Adresskonfigurationen

```
{
  "ipv6": {
    "address_configurations": []
  }
}
```

Ersetze alle existierenden IPv4 Konfigurationen durch DHCP

```
{
  "ipv4": {
    "address_configurations": [
      {
        "method": "dhcp"
      }
    ]
  }
}
```

Lege statische und dynamische IPv4 Konfiguration fest

```
{
  "ipv4": {
    "address_configurations": [
      {
        "method": "dhcp"
      },
      {
        "method": "static",
        "address": "192.168.0.100/24"
      }
    ]
  }
}
```

### 1.3.100 Netzwerkschnittstellename

Eindeutiger Name der die Netzwerkschnittstelle beschreibt.

#### Typinformationen

[NetworkInterfaceName](#) (string), pattern: `^[a-z0-9-]+$`

#### Beispiele

eth0

### 1.3.101 Statische Daten der Netzwerkschnittstelle

#### Eigenschaften

<b>iface</b> <small><a href="#">NetworkInterfaceName</a> (string), pattern: <code>^[a-z0-9-]+\$</code>, <b>required</b>, read-only</small>	Eindeutiger Name welcher Netzwerkschnittstelle beschreibt
<b>hardware_address</b> <small><a href="#">MacAddress</a> (string), pattern: <code>^([a-f0-9]{2}-){5}[a-f0-9]{2}\$</code>, <b>required</b>, read-only</small>	Eindeutige Hardwareadresse einer Netzwerkschnittstelle
<b>has_link</b> <small>boolean, <b>required</b>, read-only</small>	Aktueller physikalischer Verbindungsstatus (ob ein Kabel eingesteckt ist oder nicht)

### 1.3.102 Normierungskonstante

Normierungskonstanten sind verbunden mit der Weißreferenz.

Ihr Inhalt unterliegt dem internen Gebrauch. Somit sollte nicht direkt darauf zugegriffen werden, sondern nur durch die verbundenen API-Endpunkte (z.B. `/api/sensor/detection-profiles/{itemId}/white-reference`).

#### Typinformationen

Array of number  
colorSENSOR CFO

### 1.3.103 Freigabekanal

Beschreibt die Art der Herausgabe.

Veröffentlichungen auf dem `stable` Kanal, gelten generell als gut getestet und werden für die Nutzung in der Produktion empfohlen.

Veröffentlichungen auf dem `feature` Kanal, fügen neue Funktionen hinzu aber wurden nicht so oft wie `stable` Veröffentlichungen getestet. Feature Veröffentlichungen können, aber sollten nur mit reichlicher Überlegung in der Produktion genutzt werden.

#### Typinformationen

[ReleaseChannel](#) (string), one of [`stable`, `feature`], default: `stable`

#### Beispiele

```
stable
```

### 1.3.104 Abtastrate

#### Typinformationen

[SampleRate](#) (number), minimum: 0.01

### 1.3.105 Stichprobeneinstellungen

Stichprobeneinstellungen beschreiben alle Details der Stichprobenentnahme.

Ihre Attribute können abgefragt und untersucht werden (z.B. um die aktuelle Abtastrate abzurufen).

Die meisten Werte die in den Stichprobeneinstellungen gespeichert sind sollten nicht direkt geändert werden. Der verbundene API Endpunkt `/api/sensor/detection-profiles/current/autogain` sollte stattdessen benutzt werden.

Das einzige veränderbare Attribut innerhalb der Stichprobeneinstellungen ist der *Durchschnittswert*. Es ist sicher ihn zu ändern, obwohl die berechneten Standardwerte während eines Autogain-Betriebs optimal für die meisten Erkennungsaufgaben sein sollten.

#### Eigenschaften

<b>led_intensity</b> Number, minimum: 0, maximum: 1, <b>required</b>	Relative Intensität des Emitters während der Leuchtphase
<b>base_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Die Basisabtastrate bestimmt die Dauer der Abtastperiode. Nach jeder Abtastperiode, werden die erfassten Daten verarbeitet und ein neues Erfassungsergebnis wird berechnet (z.B. die passendste Farbgruppe (Matcher) für das gegebene Beispiel). Dies könnte den Zustand der Schaltausgänge beeinträchtigen oder konfigurierte Aktionen auslösen. Somit definiert die Basisabtastrate die Maximumrate der Änderungen für die Schaltausgänge. Siehe ebenso die effektive Abtastrate.
<b>effective_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Die effektive Abtastrate ist ein numerisches Produkt der Basisabtastrate und der Anzahl der Durchschnitte. Sie bestimmt die Minimumdauer die ein Ziel benötigt um abgetastet zu werden um dessen visuellen Auftritt korrekt zu bestimmen. Mit dem Einstellen der Standardwerte auf eins, ist dieser Wert gleich zu der Basisabtastrate.
<b>minimum_wanted_sample_rate</b> SampleRate (number), minimum: 0.01, <b>required</b>	Dieser informative Wert repräsentiert die Abtastrate, welche während des aktuellsten Autogain-Vorgangs angefragt wurde. Die effektive Abtastrate könnte von der gewollten Abtastrate abweichen, falls die angefragte Abtastrate aufgrund der Grenzen des Sensors nicht erreichbar war (z.B. Überschreiten der unterstützten Abtastrate) oder aufgrund der Umwelt (z.B. nicht genug Licht, sodass eine langsamere Amplifikation mit höherer Verstärkung nötig war).
<b>sample_light_phase</b>	Definiert ob der Sensor die internen Emmitter zum Abtasten periodisch aktivieren soll

Boolean, <b>required</b>	
<b>sample_dark_phase</b> Boolean, <b>required</b>	Definiert ob der Sensor die internen Emitter zum Abtasten periodisch deaktivieren soll.
<b>averages</b> AverageSampleCount (integer), minimum: 1, <b>required</b>	Anzahl vorheriger Stichproben die für jedes Stichprobenergebnis gemittelt werden sollen. Ein rollierender Mittelungsalgorithmus wird an den Stichproben angewandt.
<b>amplification</b> AmplificationLevel (integer), <b>required</b>	Der Amplifikationslevel gibt die interne Konfiguration eines Verstärkers an. Der Wert ist nicht dazu gedacht von normalen Nutzern manipuliert zu werden. Er soll bedient werden wie er ist (gespeichert, übermittelt und angewandt ohne Änderung oder Selbstprüfung).

### Beispiele

```
{
  "led_intensity": 0.7,
  "base_sample_rate": 1000,
  "effective_sample_rate": 1000,
  "minimum_wanted_sample_rate": 1000,
  "sample_light_phase": true,
  "sample_dark_phase": true,
  "averages": 1,
  "amplification": 5
}
```

### 1.3.106 Sensor-Fähigkeiten

Bietet den Zugang zu den von diesem Gerät unterstützten sensorischen Details (z.B. Farbräume, Eingangs- und Ausgangsleitungen, ...).

### Eigenschaften

<b>maximum_sample_rate</b> Integer, <b>required</b>	Die maximale Abtastrate die der Sensor unterstützt		
<b>tolerances</b> Array of ColorTolerance (union), <b>required</b>	Liste aus Toleranzspezifikationen welche vom Sensor unterstützt werden		
	InfiniteColorTolerance		
	<b>limits</b> Object, <b>required</b>	limits	
	<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via /api/sensor/capabilities .	
	SphereColorTolerance		
	<b>limits</b> Object, <b>required</b>	limits <b>radius</b> Numer, <b>required</b>	
	<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via /api/sensor/capabilities .	
	CylinderColorTolerance		
	<b>limits</b> Object, <b>required</b>	limits <b>radius</b> Number, <b>required</b> <b>half_height</b> Number, <b>required</b>	
	<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via /api/sensor/capabilities .	
	BoxColorTolerance		

	<b>limits</b> Object, <b>required</b>	<b>limits_half_edges</b> Array of number, minimum items: 3, maximum items: 3, <b>required</b>	
	<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .	
<b>output_drivers</b> Array of SwitchingOutputDriver (string), <b>required</b>	Liste der unterstützten elektrischen Ausgangstreiber.		
<b>trigger_sources</b> Array of TriggerSource, <b>required</b>	Beinhaltet die Liste verfügbarer Auslösequellen mit ihrem dazugehörigen Auslösefall. Auslösefälle können zum Ausführen bestimmter Aktionen automatisiert werden.		
	TriggerSource[]		
	<b>name</b> String, <b>required</b>	Name des Auslöseeingangs	
	<b>events</b> Array of TriggerEvent, <b>required</b>	TriggerEvent []	
		<b>name</b> TriggerEventName (string), <b>required</b>	
<b>output_pin_count</b> Integer, <b>required</b>	Anzahl verfügbarer Schaltausgangszeilen		
<b>actions</b> Array of Action, <b>required</b> , <b>Deprecated</b>	Veraltet: benutze stattdessen <code>/api/actions</code>		
	Action[]		
	<b>name</b> String, <b>required</b>	Eindeutiger Name einer Aktion	
	<b>arguments</b> Object, <b>required</b>	arguments	
<b>colorspaces</b> Array of Colorspace, <b>required</b>	Liste unterstützter Farbräume.		
	Colorspace[]		
	<b>name</b> String, <b>required</b>		
	<b>space_id</b> ColorspaceID, <b>required</b>	Eindeutiger Name eines Farbraums	
	<b>axes</b> Array of ColorspaceAxis, minimum items: 3, maximum items: 3, <b>required</b>	ColorspaceAxis[]	
		<b>id</b> String, <b>required</b>	Eindeutiger Name
		<b>label</b> String, <b>required</b>	Visuell lesbarer Name
		<b>minimum</b> Number, <b>required</b>	Niedrigster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
		<b>maximum</b> Number, <b>required</b>	Höchster erwarteter Wert einer Farbe entlang dieser Achse unter gewöhnlichen Umständen
<b>colorspace_tolerance_maps</b> Array of ColorspaceToleranceMap, <b>required</b>	Die Auswertung der Toleranzen entgegen der Positionen der Farben, hängt von dem aktuell eingestellten Farbraum ab. Zum Beispiel das Toleranzattribut „half_height“ bezieht sich auf die Helligkeitsbezogene Achse eines Farbraums (z.B. „L*“ für den		

	<p>„Lab*“ Farbraum) und wird benutzt für die Höhe der zylindrischen Toleranzform und der ersten Ecke der Box-Toleranzform.</p> <p>Die Farbton-bezogenen Attribute (z.B. „a“ und „b“ für den „Lab*“ Farbraum) werden für den „Radius“ einer Zylinder-Toleranzform und für die zweite und dritte Ecke der Box-Toleranzform benutzt.</p> <p>Die <i>colorspace_tolerance_maps</i> legen diese Beziehungen zwischen Farbräumen und Toleranzen fest.</p>		
	ColorspaceToleranceMap[]		
	<p><b>colorspace_id</b> ColorspaceID (string), <b>required</b></p>	Eindeutiger Name eines Farbraums	
	<p><b>tolerance_shape</b> ToleranceShapeName (string), <b>required</b></p>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .	
	<p><b>limits_axes_map</b> Object, <b>required</b></p>	limits_axes_map	
		<b>half_height</b> Array of string, optional	
		<b>half_edges</b> Array of string, optional	
		<b>radius</b> Array of string, optional	
<p><b>settings_categories</b> Array of string, <b>required</b></p>	Liste der Kategorien, welche während eines Imports ausgewählt werden können, um zu steuern welche Einstellungen angewandt werden sollen. Siehe die Dokumentation für die <b>POST</b> -Anfrage auf <code>/api/settings</code>		
<p><b>maximum_detectables_count</b> Integer, <b>required</b></p>	Maximale Anzahl an Farbpositionen (Detectable), welche in einem Erkennungsprofil gespeichert werden können.		
<p><b>maximum_matchers_count</b> Integer, <b>required</b></p>	Maximale Anzahl an Erkennungsergebnissen (Matcher), welche in einem Erkennungsprofil gespeichert werden können.		

### Beispiele

```
{
  "output_pin_count": 8,
  "tolerances": [
    {
      "shape": "infinite",
      "limits": {}
    },
    {
      "shape": "sphere",
      "limits": {
        "radius": 2
      }
    }
  ],
  {
    "shape": "cylinder",
```

```
"limits": {
  "half_height": 4,
  "radius": 2
}
},
{
  "shape": "box",
  "limits": {
    "half_edges": [
      4,
      2,
      2
    ]
  }
}
],
"actions": [
  {
    "name": "enable_switching_output",
    "arguments": {}
  },
  {
    "name": "teach_single",
    "arguments": {}
  }
],
"maximum_sample_rate": 20000,
"maximum_detectables_count": 256,
"maximum_matchers_count": 256,
"trigger_sources": [
  {
    "name": "trigger_0",
    "events": [
      {
        "name": "trigger_0_level_high"
      },
      {
        "name": "trigger_0_level_low"
      },
      {
        "name": "trigger_0_edge_rising"
      },
      {
        "name": "trigger_0_edge_falling"
      }
    ]
  },
  {
    "name": "trigger_1",
    "events": [
      {
        "name": "trigger_1_level_high"
      },
      {
        "name": "trigger_1_level_low"
      },
      {
        "name": "trigger_1_edge_rising"
      },
      {
        "name": "trigger_1_edge_falling"
      }
    ]
  }
]
```

```
]
},
{
  "name": "trigger_2",
  "events": [
    {
      "name": "trigger_2_level_high"
    },
    {
      "name": "trigger_2_level_low"
    },
    {
      "name": "trigger_2_edge_rising"
    },
    {
      "name": "trigger_2_edge_falling"
    }
  ]
},
{
  "name": "trigger_3",
  "events": [
    {
      "name": "trigger_3_level_high"
    },
    {
      "name": "trigger_3_level_low"
    },
    {
      "name": "trigger_3_edge_rising"
    },
    {
      "name": "trigger_3_edge_falling"
    }
  ]
}
],
"colorspaces": [
  {
    "axes": [
      {
        "id": "L",
        "label": "L*",
        "minimum": 0,
        "maximum": 100
      },
      {
        "id": "a",
        "label": "a*",
        "minimum": -500,
        "maximum": 500
      },
      {
        "id": "b",
        "label": "b*",
        "minimum": -200,
        "maximum": 200
      }
    ],
    "name": "L*a*b*",
    "space_id": "Lab"
  },

```

```
{
  "axes": [
    {
      "id": "L",
      "label": "L*",
      "minimum": 0,
      "maximum": 100
    },
    {
      "id": "u",
      "label": "u*",
      "minimum": 0,
      "maximum": 100
    },
    {
      "id": "v",
      "label": "v*",
      "minimum": 0,
      "maximum": 100
    }
  ],
  "name": "L*u*v*",
  "space_id": "Luv"
},
{
  "axes": [
    {
      "id": "X",
      "label": "X",
      "minimum": 0,
      "maximum": 120
    },
    {
      "id": "Y",
      "label": "Y",
      "minimum": 0,
      "maximum": 100
    },
    {
      "id": "Z",
      "label": "Z",
      "minimum": 0,
      "maximum": 120
    }
  ],
  "name": "XYZ",
  "space_id": "XYZ"
},
{
  "axes": [
    {
      "id": "x",
      "label": "x",
      "minimum": 0,
      "maximum": 1
    },
    {
      "id": "y",
      "label": "y",
      "minimum": 0,
      "maximum": 1
    }
  ],
```

```

    {
      "id": "Y",
      "label": "Y",
      "minimum": 0,
      "maximum": 100
    }
  ],
  "name": "xyY",
  "space_id": "xyY"
},
{
  "axes": [
    {
      "id": "L",
      "label": "L*",
      "minimum": 0,
      "maximum": 100
    },
    {
      "id": "u",
      "label": "u'",
      "minimum": 0,
      "maximum": 1
    },
    {
      "id": "v",
      "label": "v'",
      "minimum": 0,
      "maximum": 1
    }
  ],
  "name": "L*u'v'",
  "space_id": "uvL"
}
],
"output_drivers": [
  "off",
  "nnp",
  "pnp",
  "push-pull"
],
"colorspace_tolerance_maps": [
  {"colorspace_id": "Lab", "tolerance_shape": "box",
   "limits_axes_map": {"half_edges": ["L", "a", "b"]}},
  {"colorspace_id": "Lab", "tolerance_shape": "cylinder",
   "limits_axes_map": {"half_height": ["L"], "radius": ["a", "b"]}},
  {"colorspace_id": "Luv", "tolerance_shape": "box",
   "limits_axes_map": {"half_edges": ["L", "u", "v"]}},
  {"colorspace_id": "Luv", "tolerance_shape": "box",
   "limits_axes_map": {"half_height": ["L"], "radius": ["u", "v"]}}
],
"settings_categories": [
  "access",
  "defaults",
  "emitters",
  "firmware",
  "keypad",
  "network",
  "outputs",
  "peripherals",
  "sensor",
  "system"
]

```

```
]
}
```

### 1.3.107 Serien-Eliza-Einstellungen

#### Eigenschaften

<b>type</b> string, one of [none, eliza, modbus], default: eliza, <b>required</b>
--

### 1.3.108 Serien-Modbus-Einstellungen

#### Eigenschaften

<b>type</b> string, one of [none, eliza, modbus], default: eliza, <b>required</b>
<b>slave_id</b> any of number or null, <b>required</b>
<b>frame_format</b> string, one of [rtu, ascii], default: rtu, <b>required</b>

### 1.3.109 Signalfarbe

Ein benutzerdefinierter Farbname. Wie und welche Farbe angezeigt wird, wird vom Kunden festgelegt.

#### Typinformationen

Any of string or null

### 1.3.110 Kugel-Farbtoleranz

#### Eigenschaften

<b>limits</b> Object, <b>required</b>	limits
	<b>radius</b> number, <b>required</b>
<b>shape</b> ToleranceShapeName (string), <b>required</b>	Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via <code>/api/sensor/capabilities</code> .

### 1.3.111 Unterstützte Zeitzonen

Liste von Zeitzonen, die vom Gerät unterstützt werden.

#### Typinformationen

Array of string

#### Beispiele

```
[
  "Africa/Casablanca",
  "Antarctica/Troll",
  "Europe/Berlin",
  "UTC"
]
```

### 1.3.112 Schaltausgangstreiber

Der Ausgangstreiber definiert das elektrische Verhalten der Schaltausgänge. Die unterstützten Ausgangstreiber können über `/api/sensor/capabilities` abgerufen werden.

#### Typinformationen

[SwitchingOutputDriver](#) (string)

colorSENSOR CFO

## Beispiele

push-pull

### 1.3.113 Schaltausgänge

Elektrische Ausgabezeilen können externe Aktoren in verschiedenen elektrischen Modi ansteuern.

#### Eigenschaften

<b>output_driver</b> SwitchingOutputDriver (string), required	Der Ausgangstreiber definiert das elektrische Verhalten der Schaltausgänge. Die unterstützten Ausgangstreiber können über <code>/api/sensor/capabilities</code> abgerufen werden.
<b>count</b> integer, required	Anzahl verfügbarer Ausgabezeilen

## Beispiele

```
{
  "count": 8,
  "output_driver": "push-pull"
}
```

### 1.3.114 Schaltausgänge beschreibbar

Elektrische Ausgabezeilen können externe Aktoren in verschiedenen elektrischen Modi ansteuern.

#### Eigenschaften

<b>output_driver</b> SwitchingOutputDriver (string), required	Der Ausgangstreiber definiert das elektrische Verhalten der Schaltausgänge. Die unterstützten Ausgangstreiber können über <code>/api/sensor/capabilities</code> abgerufen werden.
--	---

### 1.3.115 Systemeinstellungen

#### Eigenschaften

<b>hostname</b> Hostname, pattern: ^(?:[a-zA-Z0-9](?:[a-zA-Z0-9\-\_]*[a-zA-Z0-9])?\.)*[a-zA-Z0-9](?:[a-zA-Z0-9\-\_]*[a-zA-Z0-9])?\$, optional	Visuell lesbarer Name, welcher den das Gerät im Netzwerk identifiziert.
<b>uptime</b> any of number or null, optional, read-only	Die aktuelle Systemverfügbarkeit in Sekunden. Auch wenn es sehr unwahrscheinlich ist, kann der Wert Null sein, wenn das System einen ungültigen Wert gemeldet hat.

## Beispiele

```
{
  "hostname": "cfo-7454232361"
}
```

### 1.3.116 Systemzeiteinstellungen

#### Eigenschaften

<b>now</b> Timestamp (string), optional	Aktuelle Zeit aus der Sicht des Sensors
<b>timezone</b> String, optional	Aktuell konfigurierte Zeitzone
<b>ntp_servers</b> Array of string, optional	Ein oder mehrere Netzwerk-Zeitserver
<b>default_ntp_servers</b> Array of string, optional, read-only	Vorkonfigurierte Netzwerk-Zeitserver

**Beispiele**

```
{
  "now": "2018-01-24T15:45:15.694004+01:00",
  "timezone": "Europe/Berlin",
  "ntp_servers": [
    "pool.ntp.org"
  ],
  "default_ntp_servers": [
    "pool.ntp.org"
  ]
}
```

**1.3.117 Zeitstempel**

Zeitstempel (Format: ISO 8601)

**Typinformationen**

[Timestamp](#) (string)

**Beispiele**

2018-01-24T14:04:26+01:00

**1.3.118 Zeitstempel Backend-Uptime**

Der Zeitstempel (in Mikrosekunden angegeben) basiert auf der Betriebszeit des internen Backends des analogen Sensors. Unter bestimmten Bedingungen kann es auf Null zurückgesetzt werden.

**Typinformationen**

[TimestampBackendUptime](#) (number), minimum: 0

**1.3.119 Toleranzformname**

Name der geometrischen Form der Toleranz. Die unterstützte Toleranzform kann abgerufen werden via `/api/sensor/capabilities`.

**Typinformationen**

[ToleranceShapeName](#) (string)

**1.3.120 Transformierte Farbe**

Eine Farbe, die durch eine Koordinate im Farbraum dargestellt wird. Die Array-Indizes der Eigenschaft `values` stimmen mit der Reihenfolge der Eigenschaft `colorspace.axes` des aktuell verwendeten Erkennungsprofils überein

**1.3.121 Auslöseereignis**

Auslöseereignisse können von ihrer Trigger-Quelle emittiert werden. An ein Auslöseereignis können Aktionen angehängt werden (siehe `/api/sensor/action-triggers`).

**Eigenschaften**

<b>name</b> TriggerEventName (string), required
--

**1.3.122 Triggerereignisname****Typinformationen**

colorSENSOR CFO

[TriggerEventName](#) (string)

**Beispiele**

trigger\_0\_level\_high

**1.3.123 Auslösequelle**

Jede Auslösequelle ist ein peripherer Eingang mit der Fähigkeit, ein oder mehrere Auslöseereignisse auszusenden.

**Eigenschaften**

<b>name</b> String, required	Name des Auslöseeingangs
<b>events</b> Array of TriggerEvent, required	TriggerEvent []
	<b>name</b> TriggerEventName (string), required

**Beispiele**

```
{
  "name": "trigger_0",
  "events": [
    {
      "name": "trigger_0_level_high"
    },
    {
      "name": "trigger_0_level_low"
    },
    {
      "name": "trigger_0_edge_rising"
    },
    {
      "name": "trigger_0_edge_falling"
    }
  ]
}
```

**1.3.124 Zustand der Auslösequelle**

Der Sensor hat eine Anzahl von Eingabezeilen welche als Auslösequelle genutzt werden können. Die Ereigniszähler werden periodisch (näherungsweise jede Sekunde) aktualisiert.

**Eigenschaften**

<b>trigger_sources</b> Array of object, required	object[]	
	<b>name</b> string, required	
	<b>event_counters</b> Object, required	event_counters
		<b>edge_falling</b> Number, required
		<b>edge_rising</b> Number, required
		<b>level_low</b> Number, required

**Beispiele**

```
{
  colorSENSOR CFO
```

```

"trigger_sources": [
  {
    "name": "trigger_0",
    "event_counters": {
      "edge_falling": 22,
      "edge_rising": 23,
      "level_low": 35124823,
      "level_high": 15
    }
  },
  {
    "name": "trigger_1",
    "event_counters": {
      "edge_falling": 0,
      "edge_rising": 0,
      "level_low": 35124832,
      "level_high": 0
    }
  }
]
}

```

### 1.3.125 Benutzer

#### Eigenschaften

<b>name</b> string, pattern: <code>^[\w-]+\$</code> , <b>required</b> , read-only	Eindeutiger Name, der ein Konto identifiziert
<b>password</b> string, optional	Diesem Konto zugewiesenes Passwort (nur beschreibbar; wird in Antworten nie zurückgegeben). Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein <code>password</code> oder ein <code>password_hash</code> angegeben werden.
<b>password_hash</b> <code>HashDigest</code> (string), pattern: <code>^[a-f0-9]+\$</code> , optional	Diesem Konto zugewiesener Passwort-Hash. Beim Anlegen eines neuen Benutzers oder beim Ändern eines Passworts muss entweder ein <code>password</code> oder ein <code>password_hash</code> angegeben werden.
<b>roles</b> Array of string, optional	Die einem Konto zugewiesenen Rollen, definieren den Berechtigungssatz des Kontos

#### Beispiele

```

{
  "name": "alice"
}

```

### 1.3.126 UUID

Eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8.

#### Typinformationen

[UUID](#) (string), pattern: `^[a-f0-9-]+$`

#### Beispiele

a014e415-0fec-4734-ac3f-30da0a5f3899

### 1.3.127 Gewünschter Schaltausgangszustand

Die Kombination von tristate Werten beschreibt einen logischen Zustand der Schaltausgänge des Sensors.

Die Zustände `true` oder `false` veranlassen den Ausgang anzusteigen oder zu sinken. Der Zustand `null` lässt den vorherigen Zustand der Ausgabe unverändert.

**Eigenschaften**

<b>uuid</b> UUID (string), pattern: <code>^[a-f0-9-]+\$, required, read-only</code>	eindeutige Kennung (UUID) gemäß RFC 4122, ITU-T Rec. X.667 und ISO / IEC 9834-8
<b>states</b> Array of any of boolean or null, <b>required</b>	Liste von True/False/Null Werten beschreibt den gesuchten Zustand der Schaltausgänge.

**Beispiele**

```
{
  "uuid": "3f26aff4-8650-42a0-b319-51776c443fbc",
  "states": [
    true,
    false,
    false,
    false,
    false,
    false,
    false,
    false
  ]
}
```

**1.3.128 Weißreferenz**

Das Weißreferenz-Attribut wird zur Angabe eines benutzerdefinierten Farbausgleichs verwendet. Sein Inhalt unterliegt der internen Verwendung. Daher sollte auf ihn nicht direkt zugegriffen werden, sondern nur über die zugehörigen API-Endpunkte (z.B. `/api/sensor/detection-profiles/{itemId}/white-reference`).

**Typinformationen**

Array of number

**1.3.129 Netzwerkschnittstellenadresse IPv4**

Netzwerkadresse (IPv4) in CIDR-Notation.

**Eigenschaften**

<b>address</b> <code>NetworkInterfaceAddressIPv4</code> (string), <b>required</b>	IPv4 Netzwerkadresse in CIDR-Notation
--	---------------------------------------

**1.3.130 Netzwerkschnittstellenadresse IPv6**

Netzwerkadresse (IPv6) in CIDR-Notation.

**Eigenschaften**

<b>address</b> <code>NetworkInterfaceAddressIPv6</code> (string), <b>required</b>	IPv6 Netzwerkadresse in CIDR-Notation
--	---------------------------------------

## 2 RS232 bzw. USB

Der textbasierte Terminal ist verfügbar über die folgenden Schnittstellen des Sensors:

- RS-232 (YSY Anschlussstifte)
- USB (optional)

### 2.1 Vorwort

#### 2.1.1 Verbindungsdetails

Der Sensor nutzt die folgende Konfiguration, um über die Serien-Schnittstelle zu kommunizieren:

<b>Baud-Rate</b>	19200 (RS-232)
<b>Datenbits</b>	8
<b>Stoppbits</b>	1
<b>Parität</b>	None
<b>Zeilenvorschub</b>	0x0A/LF/\n
<b>Verschlüsselung</b>	UTF-8

#### 2.1.2 Syntax

Die Serienkonsole verwendet Farbe, Markup und Schriftstärke, um zu verdeutlichen wie Befehle verwendet werden können und sollten. Da nicht alle Terminals Farb- und Schriftformatierungsregeln unterstützen, sehen Sie in Ihrem Terminal möglicherweise nur das beschriebene Markup.

Die Ausgabe wird wie folgt formatiert:

- Kleinbuchstaben kennzeichnen ein Schlüsselwort
- Großbuchstaben kennzeichnen eine Variable
- eckige Klammern kennzeichnen ein optionales Schlüsselwort oder Variable
- die Zeichenfolge [ . . . ] bezeichnet eine unbegrenzte Anzahl von Variablen
- optionale Schlüsselwörter und Variablen sind grau eingefärbt
- benötigte Variablen sind fett und weiß

Bitte beachten Sie bei der Eingabe die Groß- und Kleinschreibung. `matcher select` ist nicht dasselbe wie `MATCHER SELECT`.

BEISPIEL

```
matcher select MATCHER set tolerance SHAPE [limits]
```

- Toleranz einer Farbgruppe (Matcher) einstellen

Im obigen Beispiel drückt die Syntax die folgende Befehlsstruktur aus:

1. ein gefordertes Schlüsselwort „matcher“
2. ein gefordertes Schlüsselwort „select“
3. eine geforderte Variable „MATCHER“

4. ein gefordertes Schlüsselwort „set“
5. ein gefordertes Schlüsselwort „tolerance“
6. eine geforderte Variable „SHAPE“
7. eine optionale Variable „LIMITS“

### 2.1.3 Ausgabeformat & Nachrichten-Parsing

Bei Verwendung in der Automatisierung wird empfohlen, das JSON-Ausgabeformat zu verwenden. Sie können zwischen den Ausgabeformaten mit dem Befehl `set output-format` umschalten. Die verfügbaren Formate sind `json` und `human`. Bitte beachten Sie, dass das `human` Ausgabeformat Änderungen unterworfen ist und derzeit nicht für das Parsing empfohlen wird.

Sobald ein Befehl ausgeführt wird, gibt er ein oder mehrere Datenpakete aus. Unabhängig vom Ausgabeformat enden diese Pakete mit einer bestimmten Zwei-Byte-Sequenz. Das erste Byte gibt an, ob das Paket durch einen erfolgreichen Befehl erzeugt wurde und ist entweder `0x20` im Erfolgsfall oder `0x07` im Fehlerfall. Das zweite Byte kennzeichnet das Ende des Pakets und ist garantiert `0x00`, das ASCII NUL Zeichen.

### 2.1.4 Nomenklatur

Die serielle Schnittstelle verwendet eine Reihe von Namen und Bezeichnern, die eng an die REST-API des Sensors angelehnt sind.

#### UUID

UUID ist die Abkürzung für „Universally unique identifier“ und ist eine 36 Zeichen lange Zeichenfolge mit 5 alphanumerischen Gruppen, die durch Bindestriche getrennt sind. Die meisten Sammlungen verwenden einen UUID als eindeutigen Schlüssel für ihre Objekte. Die von der seriellen Schnittstelle und der zugrunde liegenden REST-API verwendeten UUIDs sind UUIDv4, wie in [RFC 4122](#) angegeben.

#### Farbposition (Detectable position)

Die Befehle zum Hinzufügen und Bearbeiten einer Farbe verwenden einen Parameter namens the position.

Dieser Wert bezieht sich auf die Farbeigenschaft in der REST-API und beschreibt die dreidimensionale Position der Farbe im aktuell aktivierten Farbraum. Das Format Regex ist

`^(\\d+(?:\\.\\d*)?),(\\d+(?:\\.\\d*)?),(\\d+(?:\\.\\d*)?)$` (Beispiel: `3.14,7.6,8`).

### 2.1.5 Typische Muster

Einige Aufgaben in der seriellen Schnittstelle haben gemeinsame oder ähnliche Verhaltensweisen. Diese wiederkehrenden Muster werden hier skizziert.

#### Sammlungen und das `select` Schlüsselwort

Sammlungen sind eine Reihe von Elementen eines bestimmten Typs, die vom Sensor gesteuert werden. Die meisten vom Sensor verwalteten Sammlungen verwenden eine UUID als eindeutigen Schlüssel, über den sie referenziert werden können.

Im Zusammenhang mit der seriellen Schnittstelle können Sammlungen leicht durch die Verwendung des Schlüsselworts „select“, gefolgt von einer Variablen als Teil der Befehlssyntax, erkannt werden. Auch wenn Sammlungen normalerweise eine UUID oder einen anderen eindeutigen Schlüssel als Bezeichner verwenden, kann die Variable andere Argumente annehmen. Sie können auch einen Listenindex verwenden, wie er durch den entsprechenden Listenbefehl für diese Sammlung definiert ist, oder den Spezialindex `-1`, um auf das letzte von Ihnen erstellte Element dieses Typs zu verweisen.

#### Eigenschaftsbefehle mit `show [PROPERTY]`

Die meisten Befehle, die detaillierte Informationen zu einem Objekt (sei es ein Sammlungselement oder eine andere Quelle) mit dem Schlüsselwort `show` anzeigen, unterstützen die Ausgabe jedes der Attribute einzeln. Der Hilfe-Befehl für den Befehl zeigt Ihnen die zulässigen Auswahlmöglichkeiten für den Eigenschaftswert an.

### 2.1.6 Unterschiede im Vergleich zu der REST-API

Das serielle Protokoll basiert auf der REST-API, aber es gibt einige Unterschiede, die in diesem Abschnitt erläutert werden.

- Es gibt keine Funktionsparität. Die REST-API ist die primäre Konfigurationsmethode für den Sensor. Nicht alle in der REST-API implementierten Funktionen sind über die serielle Schnittstelle verfügbar.
- Einige Befehle wie z.B. `samples stream` implementieren Parameter, die nicht mit dem entsprechenden API-Endpunkt übereinstimmen. Dies kann der Fall sein, wenn das durch diese Parameter gesteuerte Verhalten spezifisch für das HTTP- oder serielle Protokoll ist.
- Während die REST-API plurale Quellennamen wie Farbgruppen (`Matchers`) verwendet, verwendet das serielle Protokoll singuläre Befehlsnamen (in diesem Fall Farbgruppe (`Matcher`)).
- Es ist keine allgemeine `/defaults` API-Endpunkt-Implementierung auf der seriellen Schnittstelle verfügbar. Stattdessen implementieren Befehle wie `matcher default set hold_time` einen feinkörnigeren Zugriff.

## 2.2 Befehlsreferenz

### 2.2.1 Zugriffsbefehl

Behandle sicherheitsbezogene Aufgaben wie Logins und Sitzungsverwaltungen.

#### Unterstützte Unterbefehle

```
access login USERNAME [PASSWORD]
```

- Anmeldung mit Benutzername und Passwort für den Zugriff auf privilegierte Befehle.

```
access logout
```

- Abmeldung und Annullation der aktuellen Sitzung.

```
access session
```

- Aktuelle Sitzungsinformationen anzeigen.

### 2.2.2 Gerätebefehl

#### Unterstützte Unterbefehle

```
device [show] [PROPERTY]
```

```
[PROPERTY]: id / model_key / model_name / variant / vendor_key /  
vendor_name
```

### 2.2.3 Firmware-Befehl

Die Sensor-Firmware stellt die Konnektivität des Sensors und seiner Dienste sowie die Funktionalität des Sensor-Backends zur Verfügung.

#### Unterstützte Unterbefehle

```
firmware [version]
```

- Die aktuelle Firmware-Version anzeigen.

```
firmware recovery [show] [PROPERTY]
```

- Informationen zur Wiederherstellungs-Firmware anzeigen.

```
[PROPERTY]: channel / created_on / id / name / version
```

```
firmware recovery restore
```

- Stellen Sie das System aus dem Wiederherstellungs-Image wieder her. Alle Einstellungen werden auf ihre Standards zurückgesetzt.

```
firmware recovery upgrade
```

- Speichern Sie das aktuell laufende Firmware-Image als ein Wiederherstellungs-Image. Sie können diesen Vorgang verwenden, nachdem Sie ein erfolgreiches Firmware-Upgrade verifiziert haben.

### 2.2.4 Hilfsbefehl

Liste alle verfügbaren Befehle auf und beschreibe sie.

#### Unterstützte Unterbefehle

```
help [COMMAND]
```

```
colorSENSOR CFO
```

### 2.2.5 Tastaturbefehl

Die Tastatur bietet lokalen Zugriff auf die meisten grundlegenden Sensoraktionen.

#### Unterstützte Unterbefehle

```
keypad [show] [PROPERTY]
```

- Den Tastaturstatus anzeigen.  
[PROPERTY]: lock

```
keypad lock [STATE]
```

- Den Zustand des Tastatursperrmechanismus ändern.

### 2.2.6 Farbgruppenbefehl

Eine Farbgruppe (Matcher) bestimmt das Sensorverhalten, basierend auf dem Sensoreingang.

#### Unterstützte Unterbefehle

##### Matcher (Farbgruppe)

Eine Farbgruppe (Matcher) enthält mehrere Farben (Detectables) und das gewünschte Sensorverhalten (z.B. Ausgangszustand und Haltezeit), welches angewandt werden soll wenn eine der Farben erkannt wird.

```
matcher [list]
```

- Die Liste der konfigurierten Farbgruppen (Matchers) für das Erkennungsprofil anzeigen.

```
matcher add [OUTPUT_PATTERN]
```

- Dem Erkennungsprofil eine Farbgruppe (Matcher) hinzufügen

```
matcher remove all
```

- Alle Farbgruppen (Matchers) entfernen

```
matcher select MATCHER [show] [PROPERTY]
```

- Detaillierte Informationen über die Farbgruppe (Matcher) anzeigen  
[PROPERTY]: hold\_time | name | num\_detectables | output\_pattern  
| reset\_output\_after\_hold\_time\_expired | signal\_color | tolerance | uuid

```
matcher select MATCHER remove
```

- Eine einzige Farbgruppe (Matcher) entfernen

```
matcher select MATCHER set name NAME
```

- Den Namen einer Farbgruppe (Matcher) festlegen

```
matcher select MATCHER set hold_time DURATION
```

- Die Haltezeit einer Farbgruppe (Matcher) festlegen

```
matcher select MATCHER set output_pattern BITMASK
```

- Die Ausgabe-Bitmask einer Farbgruppe (Matcher) festlegen

```
matcher select MATCHER set tolerance SHAPE [LIMITS]
```

- Die Toleranz einer Farbgruppe (Matcher) festlegen  
SHAPE: Eine von der API definierten Formen.  
box / cylinder / infinite / sphere  
  
[LIMITS]: Ein String, der die Grenzen der Toleranz beschreibt (z.B. "2r" für eine Kugel, "4h/6r" für einen Zylinder oder "1/2/3" für eine Box). Optional für unendlich, aber erforderlich für jede andere Form.

### Matcher Defaults (Farbgruppenstandards)

Immer wenn eine neue Farbgruppe (Matcher) erstellt wird, werden einige Eigenschaften auf vordefinierte Standardwerte gesetzt. Sie können diese Standardwerte nach Ihren Wünschen ändern, um die Anzahl der anschließend erforderlichen Änderungen zu reduzieren.

```
matcher default [show] [PROPERTY]
```

- Farbgruppen-Standardwerte anzeigen  
[PROPERTY]: hold\_time | tolerance

```
matcher default set hold_time DURATION
```

- die Standard-Haltezeit für neue Farbgruppen (Matchers) festlegen

```
matcher default set tolerance SHAPE [LIMITS]
```

- die Standard-Toleranz neuer Farbgruppen (Matchers) festlegen  
SHAPE: Eine von der API definierten Formen.  
box / cylinder / infinite / sphere  
  
[LIMITS]: Ein String, der die Grenzen der Toleranz beschreibt (z.B. "2r" für eine Kugel, "4h/6r" für einen Zylinder oder "1/2/3" für eine Box). Optional für unendlich, aber erforderlich für jede andere Form.

### Farbe (Detectable)

Mehrere Farben (Detectables) können zu einer Farbgruppe (Matcher gehören). Jede Farbe (Detectable) stellt eine Position in dem aktuell aktiven Farbraum dar.

```
matcher select MATCHER detectable [list]
```

- Alle Farben (Detectables), die zu einer Farbgruppe (Matcher) gehören auflisten.

```
matcher select MATCHER detectable remove all
```

- Alle Farben (Detectables), die zu einer Farbgruppe (Matcher) gehören entfernen.

```
matcher select MATCHER detectable add [POSITION]
```

- Eine Farbe (Detectable) einer Farbgruppe (Matcher) hinzufügen. Nehmen Sie eine Probe der aktuellen Farbe, wenn keine Position angegeben ist.  
[POSITION]: Eine Position in dem aktuellen Farbraum.  
Erwartetes Format: 23.918,6,17.29113

```
matcher select MATCHER detectable select DETECTABLE [show] [PROPERTY]
```

- detaillierte Informationen über die Farben (Detectables) anzeigen  
[PROPERTY]: matcher | position | rgb | uuid

```
matcher select MATCHER detectable select DETECTABLE remove
```

- Eine einzige Farbe (Detectable) von der Farbgruppe (Matcher) entfernen.

```
matcher select MATCHER detectable select DETECTABLE set position POSITION
```

- Die Farbposition im Farbraum ändern.  
[POSITION]: Eine Position in dem aktuellen Farbraum. Erwartetes Format:  
23.918,6,17.29113

### 2.2.7 Netzwerkbefehl

Die Netzwerkkonfiguration erlaubt die Nutzung der Netzwerk-basierten Sensorfunktion (z.B. API oder die Web-Schnittstelle).

#### Unterstützte Unterbefehle

```
network [list]
```

- Den Verbindungsstatus und aktive Adressen aller Netzwerkschnittstellen anzeigen.

```
network reset
```

- Die Netzwerkkonfiguration auf die Werkseinstellungen zurücksetzen.

```
network select INTERFACE [show] [PROPERTY]
```

- Den Verbindungsstatus und aktive Adressen einer Netzwerkschnittstelle anzeigen.

[PROPERTY]: ipv4\_addresses / ipv4\_config / ipv6\_addresses /  
ipv6\_config / link / mac / name

```
network select INTERFACE set ipv4 dhcp
```

- Eine dynamisch zugewiesene IP-Adresse (über DHCP) für eine Netzwerkschnittstelle anfordern.

```
network select INTERFACE set ipv4 static ADDRESS [GATEWAY]
```

- Eine statische IPv4 Adresse für die Netzwerkschnittstelle bestimmen.

```
network select INTERFACE set ipv4 disabled
```

- Die IPv4-Konnektivität für die Netzwerkschnittstelle deaktivieren.

```
network select INTERFACE set ipv6 auto
```

- Die IPv6 zustandslose automatische Netzwerkkonfiguration (SLAAC) für die Schnittstelle aktivieren.

```
network select INTERFACE set ipv6 dhcp
```

- Eine dynamisch zugewiesene IP-Adresse (über DHCP) für eine Netzwerkschnittstelle anfordern.

```
network select INTERFACE set ipv6 static ADDRESS [GATEWAY]
```

- Eine statische IPv6 Adresse für die Netzwerkschnittstelle bestimmen.

```
network select INTERFACE set ipv6 disabled
```

- Die IPv6-Konnektivität für die Netzwerkschnittstelle deaktivieren.

### 2.2.8 Wiederholungsbefehl

Einen Befehl bequem mehrmals ausführen (z.B. folgende Änderung der Farbstichprobenergebnisse).

## Unterstützte Unterbefehle

```
repeat REPETITIONS DELAY [ARGUMENTS [...]]
```

- Einen einzelnen Befehl mehrere Male mit einer bestimmten Verzögerung wiederholen. Sie können die Ausführung mit CTRL-C abbrechen.  
REPETITIONS: Anzahl der Wiederholungen, 0 für unendlich  
DELAY: Verzögerung in Sekunden, 0 für keine Verzögerung

### 2.2.9 Stichprobenbefehl

Stichprobenergebnisse des Sensors anfordern.

## Unterstützte Unterbefehle

```
sample [show] [PROPERTY]
```

- Die aktuelle Farbstichprobe anzeigen.  
[PROPERTY]: *color / detection / output\_pattern / timestamp / trigger*

```
sample stream [COUNT] [FREQUENCY]
```

- Abrufen eines kontinuierlichen Stroms von Farbstichproben des Sensors  
[COUNT]: Anzahl der abzurufenden Datensätze, Standard 0 für unendlich  
[FREQUENCY]: Geschwindigkeit der Stichproben in Hertz, Standard unendlich

### 2.2.10 Sensorbefehl

Sensoreinstellungen beeinflussen die Stichprobenentnahme und Verarbeitung von Sensorsignalen. Geänderte Einstellungen können zuvor abgetastete Farben (Detectables) ungültig machen.

## Unterstützte Unterbefehle

```
sensor colorspace [show]
```

- Den aktuell konfigurierten Farbraum anzeigen.

```
sensor colorspace list
```

- Verfügbare Farbräume auflisten.

```
sensor colorspace set COLORSPACE
```

- Zu einem anderen Farbraum wechseln.

```
sensor autogain [SAMPLE_RATE] [TARGET_LEVEL]
```

- Die Autogain-Prozedur ausführen, um den Sensor der aktuellen optischen Umgebung anzupassen (Distanz, Lichtintensität, Zielauftritt).

```
sensor white-reference reset
```

- Die Weißreferenz auf die Werkseinstellungen zurücksetzen. Die Werkseinstellungen funktionieren gut mit einem häufig verwendeten Strahlengang (Faser und Optik). Nutzen Sie die Werkseinstellung, wenn kein angemessenes Weißreferenzziel verfügbar ist.

```
sensor white-reference sample
```

- Entnehmen Sie eine neue Weißreferenz aus dem aktuellen Ziel. Das Ziel sollte neutral weiß sein. Dies kann die Berechnung der absoluten Farbwerte innerhalb des gegebenen Farbraums verbessern.

### 2.2.11 SET-Befehl

#### Unterstützte Unterbefehle:

```
set echo STATE
```

- Aktivieren/Deaktivieren jeglicher Ausgabe von Eingabeaufforderungen oder geschriebener Text  
STATE: *off* / *on*

```
set output-format FORMAT
```

- das Antwortausgabeformat umschalten  
FORMAT: *human* / *json*

### 2.2.12 Systembefehl

Interagieren Sie mit dem System, das den Sensor betreibt.

#### Unterstützte Unterbefehle

```
system settings reset
```

- Alle Einstellungen auf die Werkseinstellungen zurücksetzen.

```
system hostname [show]
```

- Zeigt den Hostnamen des Systems an.

```
system hostname set HOSTNAME
```

- Den Hostnamen des Systems bestimmen.

```
system timezones [list]
```

- Alle vom System unterstützten Zeitzonen auflisten.

```
system time [show] [PROPERTY]
```

- Die aktuellen Zeitanstellungen des Systems anzeigen.  
[PROPERTY]: *now* | *timezone*

```
system time set now TIME
```

- Die aktuelle Zeit des Systems in ISO8601 Format setzen.

```
system time set timezone TIMEZONE
```

- Die Zeitzone des Systems festlegen.

```
system reboot
```

- Das Gerät neustarten.

### 3 Modbus Dokumentation

Das Modbus-Protokoll ist ausschließlich für die colorSENSOR CFOXXX(100) Option 100 verfügbar.

#### 3.1 Einführung

Das Modbus-Protokoll ist ein Single-Master-Protokoll. Der Datenaustausch erfolgt über eine serielle Schnittstelle oder über ein Netzwerk (TCP/IP). Der Sensor funktioniert als Modbus-Slave: Er antwortet auf Anfragen von einem Master.

Das Modbus-Protokoll erlaubt einen teilweisen Zugriff auf die wichtigsten Funktionen des Controllers. Intern verwendet es für alle Operationen die HTTP-basierte API des Controllers.

Die Modbus-Schnittstelle colorSENSOR unterstützt die folgenden Protokollmerkmale:

- Transport über TCP (IPv4 und IPv6)
- Transport über RS232 und USB mit RTU (Standard) oder ASCII-Format
- Serielle Baudraten: 9600, 19200 (Voreinstellung), 115200

Die Modbus-Slave-Adresse (nur für serielle Verbindungen relevant) ist konfigurierbar. Standardmäßig ist der colorSENSOR CFO nicht an eine bestimmte Adresse gebunden, sondern antwortet auf jedes Paket.

Der vollständige Satz der unterstützten Befehle ist als JSON-Dump verfügbar. Dieser strukturierte Datensatz soll die Erstellung eines herstellerspezifischen Modbus-Mappings für den Sensor erleichtern.

### 3.2 Schnellstart

Die folgenden Konfigurationsdetails und Hinweise sollen die ersten Schritte mit der Modbus-Protokollimplementierung der Sensoren erleichtern:

- Schließen Sie den Controller über RS232 (Baudrate: 19200), USB oder TCP (Port 502) an, um über das Modbus-Protokoll zu kommunizieren.
- Verwenden Sie Big-Endian (Byte-Ordnung und Wort-Ordnung), wenn Sie Daten der Modbus-Antworten interpretieren.
- Berücksichtigen Sie das 1-basierte Adressierungsschema beim Zugriff auf Register. Zum Beispiel wird eine dokumentierte Adresse von 501 über die Leitung als 500 übertragen. Die meisten Modbus-Client-Implementierungen wenden diese Übersetzung implizit an. Nur sehr wenige Implementierungen verwenden stattdessen die On-Wire-Adresse. In diesem Fall muss die dokumentierte Adresse für diese spezifischen Clients dekrementiert werden.
- Rufen Sie die Input-Register von 500 bis 508 über eine Modbus-Anfrage ab. Diese Register enthalten feste Werte in verschiedenen Formaten (z.B. Float, 32 Bit und 64 Bit Integer). Stellen Sie sicher, dass Ihre Client-Implementierung diese Werte entsprechend ihrem dokumentierten Wert korrekt interpretiert (siehe die Dokumentation zum Registerinhalt). Im Falle von Fehlinterpretationen müssen Sie möglicherweise die Bytereihenfolge oder den Adressoffset Ihrer Client-Implementierung anpassen.

### 3.3 Unterstützte colorSENSOR-Funktionen

Die Modbus-Schnittstelle der Farbsensoren bietet die meisten Funktionen der folgenden API-Endpunkte:

- /defaults (only matcher-related defaults)
- /device
- /firmware (only status retrieval; no upgrade)
- /firmware/recovery
- /firmware/recovery/upgrade-from-current
- /sensor/samples/current
- /sensor/matchers
- /sensor/detectable
- /sensor/detection-profiles
- /sensor/detection-profiles/autogain
- /sensor/detection-profiles/white-reference
- /sensor/capabilities
- /system
- /system/factory-reset
- /system/reboot
- /peripherals/outputs
- /peripherals/rs232
- /peripherals/usb
- /settings

Die folgenden API-Endpunkte werden aufgrund der Unbeständigkeit ihrer Daten oder ihrer Komplexität (die innerhalb der API schwer auszudrücken sind) im Modbus-Protokoll nicht unterstützt.

- /access
- /action-triggers
- /actions
- /firmware/images
- /firmware/settings
- /network/interfaces/
- /peripherals/keypad
- /peripherals/trigger-sources
- /system/time
- /system/time/zones

### 3.4 Datentypen und Registeradressierung

#### 3.4.1 Datentypen und Modbus-Funktionen

Das Modbus-Protokoll spezifiziert verschiedene Funktionen für den Zugriff auf und die Veränderung von Werten.

Die folgenden Funktionen (und ihre jeweiligen Funktionscodes) werden für die verschiedenen Datentypen verwendet:

Funktion	Code	Funktionsname
Bits nur lesen	2	Diskrete Eingänge lesen
Beschreibbare Bits	1	Einzelne Bits lesen
	5	Einzelne Bits schreiben
	15	Mehrfachbits schreiben
Wörter nur lesen	4	Eingabe-Register lesen
Wörter schreiben	3	Lesen mehrerer Holding Register
	6	Schreiben eines einzelnen Holding Register
	16	Schreiben mehrerer Holding
	23	Lesen/Schreiben mehrerer Register

#### 3.4.2 Register Adressen

Die Adressierung von Daten über das Modbus-Protokoll ist nicht streng spezifiziert. Verschiedene Implementierungen verwenden eine Vielzahl von Namensschemata und Offsets. Die relevanten Details dieser Modbus-Implementierung sind

- Alle in dieser Dokumentation geschriebenen Adressen sind Register-Offsets, die sich auf die spezifische Modbus-Funktion beziehen.
- Alle Adressen sind 1-basiert. Dieser Ansatz wird von den meisten Modbus-Implementierungen verwendet.

Zum Beispiel ist das Register für den Float-Testwert als schreibgeschütztes Wort an Adresse 501 dokumentiert. Diese Adresse könnte auch als 30501 geschrieben werden (basierend auf einem traditionellen Modbus-Adressierungsschema, das die Funktionen auf bestimmte Adressbereiche abbildet). Der Inhalt dieses Registers kann mit der Funktion Read Input Registers (Funktionskennung "4") abgerufen werden. Die interne Adresse dieses Wertes (wie sie für das On-Wire-Format von Modbus verwendet wird) ist 500 (aufgrund der 1-basierten Registeradressierung). Diese interne Adresse wird nur von sehr wenigen Modbus-Client-Implementierungen verwendet. Die meisten Implementierungen verwenden stattdessen die 1-basierte Adresse.

Clients ohne Unterstützung für Adress-Offsets müssen möglicherweise jede Adresse (wie hier dokumentiert) beim Zusammensetzen des Modbus-Datenrahmens dekrementieren.

#### 3.4.3 Einfache Datentypen

Die Modbus-Spezifikation beschreibt einfache Datentypen (Bits und 16-Bit-Wörter). Zusätzlich werden die folgenden Datentypen von der Modbus-Implementierung des colorSENSORS verwendet:

- Float-Werte: Zwei Register (32 Bit), IEEE-754, Big-Endian-Wortreihenfolge und Byte-Reihenfolge.
- Integer-Werte mit 32 Bit (zwei Register) oder 64 Bit (vier Register): Big-Endian-Wortreihenfolge und Byte-Reihenfolge.
- String: Das erste Wort enthält die Länge; alle folgenden Bytes enthalten die ASCII-Zeichen. Jedes "Wort"-Register (nach der Länge) enthält zwei Zeichen (erstes: oberes, zweites: unteres Byte). Lesen über das Ende der String-Länge hinaus ist erlaubt und liefert Null-Bytes. Daher ist normalerweise ein nachlaufendes Null-Byte am Ende des Strings vorhanden. Darauf darf man sich aber nicht verlassen, colorSENSOR CFO

da das nachlaufende Null-Byte fehlt, wenn der String genau die maximale Anzahl der erlaubten Zeichen für diesen String verwendet.

- Bytes: Für binäre Datenübertragungen wird ein Rohbyte-Array verwendet. Jedes Register enthält zwei Zeichen (erstes: oberes, zweites: unteres Byte). Das Lesen über das Ende der Binärdaten hinaus ist erlaubt und liefert Null-Bytes zurück. Die Länge der Rohdaten sollte über ein separates Register gehandhabt werden.
- Bitmask: 16-Bit-Wörter werden verwendet, um boolesche Felder darzustellen oder zu manipulieren. Jedes Bit repräsentiert einen einzelnen booleschen Wert. Die Beschreibung jedes Bitmask-Datenfeldes ordnet Bitpositionen dem durch dieses Bit beschriebenen booleschen Zustand zu. Ein Wert von Null wird als false (nicht aktiv) angesehen. Ein Wert von Eins ist true. Die Bitpositionen beginnen bei Null mit dem niedrigstwertigen Bit.

### 3.5 Sitzungsstatus, Aktualität und mehrere Schnittstellen

Mehrere Schnittstellen des Sensors können über das Modbus-Protokoll kommunizieren. Jede Hardware-schnittstelle (z.B. RS232, USB) verwaltet ihren eigenen Zustand. Dies ist relevant für zustandsabhängige Operationen (z.B. Zugriff auf eine Sammlung), die eine Abfolge von Lese- oder Schreibanforderungen erfordern. Die Ethernet-Schnittstelle akzeptiert TCP-Verbindungen. Jede Verbindung verfolgt ihren eigenen Status für die Dauer der Verbindung.

#### 3.5.1 Funktionen

##### 3.5.1.1 API-Endpunkt „Einstellen und aktivieren der automatischen Aussteuerung“: /sensor/detection-profiles/current/autogain

„Autogain-Prozedur“ ausführen, um geeignete Abtasteigenschaften für die aktuelle optische Umgebung zu bestimmen. Die resultierende Abtasteinrichtung wird automatisch angewendet. Diese neuen Einstellungen werden wirksam, sobald die Antwort „erfolgreich abgeschlossen“ gesendet wird. Der Erfolg oder Misserfolg einer Autogain-Prozedur kann verifiziert werden, sobald das „autogain\_is\_running“-Flag gelöscht wird.

Adresse	Datentyp	Operation	Beschreibung	FC	
00020	Bit	schreiben	Autogain starten	5, 15	
00302	Bitmask	lesen	Status des zuletzt gestarteten Autogain-Verfahrens	4	
			<b>Position</b>		<b>Beschreibung</b>
			0		läuft noch
			1		erfolgreich abgeschlossen
		2	Gescheitert: Ziel ist zu dunkel		
00410	Float	lesen / schreiben	Minimal gewünschte Abtaste	3, 4, 6, 16	
00412	Float	lesen / schreiben	Ziel-Analog-Eingangsspegel	3, 4, 6, 16	
00414	Uint16	lesen / schreiben	Anzahl der für die Mittelwertbildung verwendeten Muster	3, 4, 6, 16	
00415	Bitmask	lesen / schreiben	Digitale Merker für Autogain-Verfahren	3, 4, 6, 16	
			Standardwert: 65535		
			<b>Position</b>		<b>Beschreibung</b>
		0	Internen Ermittler aktivieren		
		1	Umgebungslichtkompensation aktivieren		
00416	Bitmask	lesen / schreiben	Standard-Autogain-Einstellungen mit benutzerdefinierten Werten überschreiben	3, 4, 6, 16	
			<b>Position</b>		<b>Beschreibung</b>
			0		Überschreiben der minimal gewünschten Abtaste
			1		Überschreiben des analogen Zieleingangspegels
		2	Die Anzahl der für die Mittelwertbildung verwendeten Proben überschreiben		

##### 3.5.1.2 API-Endpunkt „Weiß-Referenz setzen / zurücksetzen“: /Sensor/Erkennungs-Profile/Strom/Weiß-Referenz

Die Weiß-Referenz wird zur Berechnung genauer Farbpositionen in den Farbräumen verwendet. Die werkseitig voreingestellte Weiß-Referenz ist für einen bestimmten Satz von Fasern und Optiken geeignet. Eine benutzerdefinierte Weiß-Referenz kann abgetastet werden. Hierfür wird ein Weißstandard empfohlen.

Adresse	Datentyp	Operation	Beschreibung	FC
00021	Bit	schreiben	Weiß-Referenz auf Werkseinstellung zurücksetzen	5
00022	Bit	schreiben	Weiß-Referenz setzen	5

##### 3.5.1.3 API-Endpunkt „Neue Farbtabelle hinzufügen“: /sensor/matchers

Adresse	Datentyp	Operation	Beschreibung	FC
00024	Bit	schreiben	Erstellen einer neuen Farbgruppe und zuweisen der aktuelle Schaltausgangs (als detektierbar).	5,15

00451	Uint16	lesen	Abrufen der Kennung der zuletzt erstellten Farbgruppe	4
-------	--------	-------	---	---

### 3.5.1.4 API-Endpunkt „Verwalten der Farben einer Farbgruppe“: /sensor/matchers

Jede Farbgruppe (Matcher) kann sich auf eine oder mehrere Farben (Detectables) beziehen.

Adresse	Datentyp	Operation	Beschreibung	FC
00025	Bit	schreiben	Hinzufügen einer neuen erkennbaren Farbe zu einer vorhandenen Matcher (Farbgruppe)	5
00026	Bit	schreiben	Löschen aller Farben in einer Farbgruppe	5
00027	Bit	lesen	Gibt an, ob die aktuell ausgewählte Farbgruppe existiert	1
00311	Uint16	lesen	Aktuelle Anzahl der der Farbgruppe zugewiesenen Farbe (Detectables)	4
00450	Uint16	lesen / schreiben	Bestimmen der Farbgruppe zum Hinzufügen oder Entfernen von Farben	3, 6

### 3.5.1.5 API-Endpunkt "Sensorfähigkeiten lesen": /sensor/capabilities

Lesen der eingestellten Funktionen im Sensor.

Adresse	Datentyp	Operation	Beschreibung	FC	
00300	Uint16	lesen	Anzahl der verfügbaren Schaltausgänge	4	
00301	Bitmask	lesen	Vom Sensor unterstützte Farbräume	4	
			<b>Position</b>		<b>Beschreibung</b>
			0		XYZ
			1		L*a*b*
			2		xyY
			3		L*u*v*
4	L*u'v'				
00303	Bitmask	lesen	Verfügbare Toleranzformen	4	
			<b>Position</b>		<b>Beschreibung</b>
			0		Klassifizierung
			1		Kugel
			2		Zylinder
3	Box				
00304	Bitmask	lesen	Verfügbare Schaltausgangstreiber	4	
			<b>Position</b>		<b>Beschreibung</b>
			0		deaktiviert
			1		NPN
			2		PNP
3	Push-Pull				
00305	Float	lesen	Maximale Abtastrate	4	
00307	Uint16	lesen	Maximale Anzahl von Nachweisbarkeiten	4	
00308	Uint16	lesen	Maximale Anzahl von Abgleichen	4	

### 3.5.1.6 API-Endpunkt „Aktuelle Messwerte abrufen“: /sensor/samples/current

Auslesen der neuesten Farbmesswerte. Ein einziger Lesevorgang, der den gesamten Speicherbereich des Musters abdeckt, ist garantiert konsistent. Mehrere Lesevorgänge in Serie werden wahrscheinlich zu einer Kombination von Werten aus den verschiedenen Mustern führen, die in der Zeit zwischen der ersten und der letzten Anfrage gesammelt wurden.

Adresse	Datentyp	Operation	Beschreibung	FC
00150	Uint64	lesen	Zeitstempel des aktuellen Musters	4
00154	Float	lesen	Signal-Level des aktuellen Musters	4
00156	Float	lesen	Darstellung der Farbe im XYZ-Farbraum (Element X)	4
00158	Float	lesen	Darstellung der Farbe im XYZ-Farbraum (Element Y)	4
00160	Float	lesen	Darstellung der Farbe im XYZ-Farbraum (Element Z)	4
00162	Float	lesen	Darstellung der Farbe im aktuell aktiven Farbraum (L*)	4
00164	Float	lesen	Darstellung der Farbe im aktuell aktiven Farbraum (a*)	4
00166	Float	lesen	Darstellung der Farbe im aktuell aktiven Farbraum (b*)	4
00168	Float	lesen	Darstellung der Farbe als RGB-Wert (Wert R)	4
00170	Float	lesen	Darstellung der Farbe als RGB-Wert (Wert G)	4
00172	Float	lesen	Darstellung der Farbe als RGB-Wert (Wert B)	4

Adresse	Datentyp	Operation	Beschreibung	FC
00174	Uint16	lesen	Eingänge mit einem High-Level-Ereignis während der letzten Abtastperiode (Bit 0 -> IN0)	4
00175	Uint16	lesen	Eingänge mit einem Niedrig-Level-Ereignis während der letzten Abtastperiode (Bit 0 -> IN0)	4
00176	Uint16	lesen	Eingänge mit einem Ereignis mit ansteigender Flanke während der letzten Abtastperiode (Bit 0 -> IN0)	4
00177	Uint16	lesen	Eingänge mit einem Ereignis mit fallender Flanke während der letzten Abtastperiode (Bit 0 -> IN0)	4
00178	Uint16	lesen	ID der nächstgelegenen Farbgruppe im Bereich der Farbposition des letzten Musters. Der Wert 65535 wird zurückgegeben, wenn die abgetastete Farbe nicht im Bereich einer der verfügbaren Übereinstimmungen lag.	4
00179	Uint16	lesen	Derzeit aktiver Zustand der Schaltausgänge (Bit 0 -> OUT0)	4
00180	Float	lesen	Abstand (basierend auf den Achsen des aktuell konfigurierten Farbraums) zwischen der zuletzt abgetasteten Farbe und der nächstgelegenen geeigneten Farbgruppe (falls vorhanden). Ein negativer Wert (-1) zeigt an, dass sich keine Farbgruppe im Bereich befindet. 1. Distanzwert	4
00182	Float	lesen	Abstand (basierend auf den Achsen des aktuell konfigurierten Farbraums) zwischen der zuletzt abgetasteten Farbe und der nächstgelegenen geeigneten Farbgruppe (falls vorhanden). Ein negativer Wert (-1) zeigt an, dass sich keine Farbgruppe im Bereich befindet. 2. Distanzwert	4
00184	Float	lesen	Abstand (basierend auf den Achsen des aktuell konfigurierten Farbraums) zwischen der zuletzt abgetasteten Farbe und der nächstgelegenen geeigneten Farbgruppe (falls vorhanden). Ein negativer Wert (-1) zeigt an, dass sich keine Farbgruppe im Bereich befindet. 3. Distanzwert	4

### 3.5.1.7 API-Endpunkt „Status der Farbtabelle“: [/sensor/detection-profiles/current](#)

Abrufen der aktuellen Verwendung der Farbtabelle.

Adresse	Datentyp	Operation	Beschreibung	FC
00309	Uint16	lesen	Aktuelle Anzahl von Übereinstimmungen (Farbgruppen), die in der Farbtabelle gespeichert sind	4
00310	Uint16	lesen	Aktuelle Anzahl der in der Farbtabelle gespeicherten Farben	4

### 3.5.1.8 API-Endpunkt „Farbtabelle löschen“: [/sensor/matcher](#)

Löschen Sie alle Farben, die in der Farbtabelle gespeichert sind.

Adresse	Datentyp	Operation	Beschreibung	FC
00023	Bit	lesen	Alle gespeicherten Farben entfernen	5, 15

### 3.5.1.9 API-Endpunkt „Schaltausgangstreiber einstellen“: [/peripherals/outputs](#)

Elektrische Ausgangsleitungen können externe Aktoren in verschiedenen elektrischen Modi ansteuern. Der derzeit aktive Modus kann abgerufen und geändert werden.

Adresse	Datentyp	Operation	Beschreibung	FC	
00400	Uint16	lesen / schreiben	Abrufen und Ändern des Stromschaltausgangstreibers. Mögliche Werte:	3,4,6,16	
			<b>Beschreibung</b>		<b>Werte</b>
			off		0
			nnp		1
			push-pull		3

### 3.5.1.10 API-Endpunkt „Firmware-Version abfragen“: [/firmware](#)

Lesen Sie Informationen über die Firmware.

Adresse	Datentyp	Operation	Beschreibung	FC
00100	Uint16	lesen	Firmware Version (Major: X.0.0)	4
00101	Uint16	lesen	Firmware Version (Major: 0.X.0)	
00102	Uint16	lesen	Firmware Version (Major: 0.0.X)	

### 3.5.1.11 API-Endpunkt „Geräteinformation“: [/device](#)

Lesen Sie Informationen über das Gerät.

Adresse	Datentyp	Operation	Beschreibung	FC
00103	String	lesen	Geräteseriennummer	4
00114	String	lesen	Gerätehersteller	
00123	String	lesen	Gerätemodell	
00132	String	lesen	Gerätevariation	

### 3.5.1.12 API-Endpoint „Zugangsverwaltung“

Sperren oder Entsperren bestimmter Methoden für den Zugriff auf den Sensor.

Adresse	Datentyp	Operation	Beschreibung	FC	
00460	Bitmask	lesen / schreiben	Lesen und setzen der Zugangssperre für Tastatur.	3, 4, 6, 16	
			<b>Position</b>		<b>Beschreibung</b>
			0		Tastatur sperren
			1		Ablehnung des Lesezugriffs für die API
			2	Schreibzugriff für die API verweigern	

### 3.5.1.13 API-Endpoint „Benutzerverwaltung“: /access/user

Verwalten Sie die von der HTTP-API verwendeten Benutzerkonten.

Adresse	Datentyp	Operation	Beschreibung	FC
00028	Bit	schreiben	Entfernt alle API-Benutzer	1, 5, 15
00461	Uint16	lesen	Anzahl der API-Benutzer	4

### 3.5.1.14 API-Endpoint „Einstellungen zurücksetzen“: /settings

Zurücksetzen der Controllereinstellungen.

Adresse	Datentyp	Operation	Beschreibung	FC
00006	Bit	schreiben	Alle Einstellungen zurücksetzen	5

### 3.5.1.15 API-Endpoint „Auf Werkseinstellung zurückstellen“: /system-/factory-reset

Controllerfirmware auf die Werkseinstellungen zurücksetzen und neustarten.

Adresse	Datentyp	Operation	Beschreibung	FC
00002	Bit	schreiben	Auslösen eines Werksresets der Firmware und der Einstellungen.	5

### 3.5.1.16 API-Endpoint „System Neustart“: /system/reboot

Neustart aller Sensorkomponenten.

Adresse	Datentyp	Operation	Beschreibung	FC
00001	Bit	schreiben	Einen Neustart auslösen	5

### 3.5.1.17 API-Endpoint „Upgrade auf Wiederherstellungs-Firmware“: /system-/factory-reset

Gespeichertes Wiederherstellungsabbild durch die aktuelle System-Firmware ersetzen. Dies ist hilfreich, wenn Sie das Wiederherstellungsabbild auf eine neuere Firmware-Version aktualisieren wollen.

Adresse	Datentyp	Operation	Beschreibung	FC
00003	Bit	schreiben	Aktualisieren der Wiederherstellungs-Firmware auf die derzeit laufende Firmware-Version	5

### 3.5.1.18 API-Endpoint „RS232-Schnittstelle konfigurieren“: /peripherals/rs232

Überprüfen und Einstellen der RS232-Schnittstelle des Controllers . Einige Einstellungen beziehen sich auf das Modbus-Slave-Protokoll. Die Modbus-Slave-ID wird für die serielle Kommunikation verwendet, wenn mehr als ein Modbus-Gerät an denselben Bus angeschlossen ist. Das Rahmenformat kann je nach Bedarf des Modbus-Masters geändert werden.

Adresse	Datentyp	Operation	Beschreibung	FC
00430	Uint16	lesen / schreiben	Baudrate der RS232-Schnittstelle	3,4,6,16
			<b>Werte</b>	

			9600	0	
			19200	1	
			115200	2	
00431	Uint16	lesen / schreiben	Für die RS232-Schnittstelle zu verwendendes Protokoll		3,4,6,16
			<b>Werte</b>	<b>Beschreibung</b>	
			eliza	0	
			modbus	1	
00432	Uint16	lesen / schreiben	Für das Modbus-Protokoll zu verwendende Slave-ID (1..247)		3,4,6,16
00433	Uint16	lesen / schreiben	Für das Modbus-Protokoll zu verwendendes Rahmenformat. Mögliche Werte:		3,4,6,16
			<b>Werte</b>	<b>Beschreibung</b>	
			rtu	0	
			ascii	1	

### 3.5.1.19 API-Endpunkt „USB-Schnittstelle konfigurieren“: /peripherals/usb

Überprüfen und Einstellen USB-Schnittstelle des Controllers. Einige Einstellungen beziehen sich auf das Modbus-Slave-Protokoll. Die Modbus-Slave-ID wird für die serielle Kommunikation verwendet, wenn mehr als ein Modbus-Gerät an denselben Bus angeschlossen ist. Das Rahmenformat kann je nach Bedarf des Modbus-Masters geändert werden.

Adresse	Datentyp	Operation	Beschreibung	FC	
00440	Uint16	lesen / schreiben	Für die USB-Schnittstelle zu verwendendes Protokoll. Mögliche Werte:		3, 4, 6, 16
			<b>Werte</b>	<b>Beschreibung</b>	
			eliza	0	
			modbus	1	
00441	Uint16	lesen / schreiben	Für das Modbus-Protokoll zu verwendende Slave-ID (1..247)		3, 4, 6, 16
00442	uint16	lesen / schreiben	Für das Modbus-Protokoll zu verwendendes Rahmenformat. Mögliche Werte		3, 4, 6, 16
			<b>Werte</b>	<b>Beschreibung</b>	
			rtu	0	
			ascii	1	

### 3.5.1.20 Adressen zum Testen der Datenformate

Einige Register antworten mit festgelegten Festwerten, damit die Richtigkeit des konfigurierten Datenformats leicht überprüfen werden kann.

Adresse	Datentyp	Operation	Beschreibung	FC
00500	Uint16	lesen	Ein 16-Bit -Ganzzahlwert, der die Zahl 1234 enthält.	4
00501	Float	lesen	Ein Gleitkommawert, der die Zahl -1.0 enthält.	4
00503	Uint32	lesen	Ein 32-Bit-Ganzzahlwert, der die Zahl 12345678 enthält.	4
00505	Uint64	lesen	Ein 64-Bit-Ganzzahlwert, der die Zahl 123456789012 enthält.	4



MICRO-EPSILON Eltrotec GmbH  
Manfred-Wörner-Straße 101 · 73037 Göppingen / Deutschland  
Tel. +49 (0) 7161 / 98872-300 · Fax +49 (0) 7161 / 98872-303  
info@micro-epsilon.de · www.micro-epsilon.de  
Your local contact: [www.micro-epsilon.com/contact/worldwide/](http://www.micro-epsilon.com/contact/worldwide/)

X9750375.01-A021012HDR  
© MICRO-EPSILON MESSTECHNIK