

# **AVM**

# **Home Automation**

## **HTTP Interface**

24.04.20

**AVM**

## Änderungshistorie

Version	Datum	Ab FRITZ!OS	Änderung
1.0	22.04.2013		Erstes Release
1.01	26.04.2013		Beispiel-URL hinzugefügt
1.02	12.06.2013		switchcmd getswitchlist hinzugefügt
1.03	30.07.2013		Leerzeichen in AIN entfernt, ain Parameter kann AIN oder MAC sein
1.04	19.11.2013		switchcmd setswitchtoggle hinzugefügt
1.05	16.12.2013		Hinweis zur FRITZ!OS Version
1.06	4.7.2014		Switchcmd getdevicelistinfos hinzugefügt
1.07	16.9.2014		Switchcmd gettemperature hinzugefügt
1.08	3.6.2015	6.35	Hkr-Solltemperatur setzen und auslesen
1.09	5.1.2016	6.50	Ab Firmware 6.50 sid-Parameter(SessionID) immer benötigt, Session-ID Referenz korrigiert
1.10	18.4.2016	6.69	Getdevicelistinfos: Heizkörperregler mit Errorcode- und Batterylow-Info
1.11	24.5.2016	6.69	Getdevicelistinfos: Schaltsperre am Gerät in <devicelock>-Info
1.12	5.7.2016	6.69	Getdevicelistinfos: HKR nextchange-Info
1.13	30.8.2016	6.69	Getdevicelistinfos: alert-Info
1.14	11.1.2017		Getdevicelistinfos: voltage-Info
1.15	31.05.2017	6.98	Neue getbasicdevicestats Abfrage
1.16	4.10.2017	6.98	HTTPS-Port über TR-064-Service
1.17	4.12.2017	6.98	erweiterter Hinweis zu functionbitmask
1.18	29.1.2018	6.98	Mikrofon-functionbitmask
1.19	5.2.2018	6.98	Windowopenactiv-Information zum HKR
1.20	1.3.2018	6.98	HAN-FUN und HAN-FUN Unit Vorlagen/Template gettemplatelistinfos und applytemplate
1.21	20.3.2018	7.00	Vorlage/Template geändert
1.22	28.3.2018	7.00	Getdevicelistinfos: button-Info
1.23	22.01.2019	7.08	<button>-Info für FD400 erweitert, optionale Attribute hinzugefügt <battery> und <batterylow>-Info für <device>(wenn vorhanden)
1.24	18.06.2019	7.15	Color, Level und HANFUNK-Aktualisierung Getdevicelistinfos-XML um 3 Unterknoten erweitert: <simpleonoff>, <levelcontrol> und <colorcontrol> 4 neue switchcmd: setsimpleonoff, setlevel, setcolor, setcolortemperature
1.25	28.06.2019	7.15	getcolordefaults switchcmd hinzugefügt
1.26	11.10.2019	7.15	HAN-FUN Einführung <txbusy> bei getdevicelistinfos XML hinzugefügt
1.27	25.11.2019	7.15	Hinweis zu unzulässigen Farb- und Farbtemperaturwerten minimal FRITZ!OS-Version aktualisiert
1.28	5.12.2019	7.15	<levelpercentage>-Info und setlevelpercentage switchcmd hinzugefügt
1.29	14.01.2020	7.15	Hinweis zu HSV-Farbraum im HueSaturation-Mode
1.30	15.01.2020	7.15	Getcolordefaults liefert zusätzlich zu Hue und Saturation auch Value
1.31	30.01.2020	7.15	HKR-Boost und HKR-Fenster-offen Unterstützung hinzugefügt neue switchcmd sethkrboost und sethkrwindowopen <hkr>-Info um <boostactive>, <boostactiveendtime>, <windowopenactive>, <windowopenactiveendtime>, <holidayactive> und <summeractive> erweitert <lastalertchgtimestamp>-Info bei Alarmsensor
1.32	26.02.2020	7.15	<windowopenactiv>
1.33	24.04.2020	7.15	neues switchcmd setblind

# Inhaltsverzeichnis

1 Einführung.....	4
1.1 HAN-FUN.....	4
2 HTTP Methode und Session-ID.....	4
3 Kommandos und Rückgabewerte.....	5
3.1 Beispiele für switchcmd Aufrufe.....	6
3.2 XML-Format von getdevicelistinfos Kommando.....	7
3.3 XML-Format von getbasicdevicestats Kommando.....	11
3.4 XML-Format von gettemplatelistinfos Kommando.....	12
3.5 XML-Format von getcolordefaults Kommando.....	13
4 Fehlercodes.....	15

## 1 Einführung

Für einfache Schaltvorgänge bietet die Home Automation Komponente der FRITZ!Box eine HTTP Schnittstelle an. Über diese können Zustandsabfragen und Schaltvorgänge von einzelnen Aktoren durchgeführt werden.

Voraussetzung für die Nutzung der HTTP Schnittstelle ist eine FRITZ!Box mit Smart Home Unterstützung. Für die Schnittstellenversion 1.08 wird FRITZ!OS ab Version 6.35 benötigt. Für die Schnittstellenversion 1.13 wird FRITZ!OS ab Version 6.69 benötigt.

### 1.1 HAN-FUN

Die FRITZ!Box unterstützt das HAN FUN App Layer Protokoll der (DECT-)ULE Alliance. Auf avm.de finden sie eine Liste der unterstützten HAN-FUN-Geräte(siehe Unterstützung von DECT-ULE/HAN-FUN)

HAN-FUN benutzt ein Konzept von Units je HAN-FUN-Gerät. So kann es zum Beispiel mehrere Alarm-Units oder OnOff-Units je HAN-FUN-Gerät geben. Eine Unit hat einen Unit-Type(z.B. Bewegungsmelder oder dimmbare Farb-Lampe) und eine Liste von unterstützten HAN-FUN-Interfaces(z.B. Alert, OnOff, Color oder Level/Helligkeit). Die Unit hat einen eigenen Identifier, der benutzt werden muss, wenn ein (Schalt-, Helligkeits-)Befehl an die Unit gesendet werden soll.

## 2 HTTP Methode und Session-ID

Die Kommandos werden über einen HTTP GET Request an die URL

```
https://fritz.box/webservices/homeautoswitch.lua?ain=<ain>&switchcmd=<cmd>&sid=<sid>
```

abgesetzt.

Der HTTPS-Port ist auf der FRITZ!Box konfigurierbar. Er kann über den X\_AVM-DE\_RemoteAccess TR-064-Service und der dazugehörigen GetInfo-Action mit der "NewPort"-Variable abgefragt werden. Siehe TR064-Spezifikation in [1].

Dem HTTP-Request sind dazu folgende Parameter URL-encoded zu übergeben:

Parameter	Bedeutung
ain	Identifikation des Aktors oder Templates, z. B. "012340000123" oder MAC Adresse für Netzwerkgeräte
switchcmd	Auszuführendes Kommando, s. Tabelle 2: Kommandos
param	Für einige switchcmd zusätzlich benötigter Parameter, s. Tabelle 2: Kommandos
sid	Session-ID, ab FRITZ!OS 6.50 wird der sid-Parameter immer benötigt, Spezifikation in [2], (ab FRITZ!OS 6.50 wird bei der Einrichtung ein FRITZ!Box-Passwort erzwungen)

Tabelle 1: GET Parameter

Die HTTP Response enthält den zum Kommando zugehörigen Status als Text. Der Content-Type ist "text/plain; charset=utf-8".

Ausnahme bei getdevicelistinfos und getbasicdevicestats: Die HTTP Response enthält die Device-Liste als XML. Der Content-Type ist "text/xml; charset=utf-8".

Beispiel zum Einschalten des Aktors mit der AIN "012340000123" und Session-ID "9c977765016899f8":

```
https://fritz.box/webservices/homeautoswitch.lua?  
ain=012340000123&switchcmd=setswitchon&sid=9c977765016899f8
```

### 3 Kommandos und Rückgabewerte

Kommando	HTTP-Parameter	Aktion	Antwort
getswitchlist	switchcmd, sid	Liefert die kommaseparierte AIN/MAC Liste aller bekannten Steckdosen	kommaseparierte AIN/MAC-Liste, leer wenn keine Steckdose bekannt
setswitchon	switchcmd, sid	Schaltet Steckdose ein	"1"
setswitchoff	switchcmd, sid ain	Schaltet Steckdose aus	"0"
setswitchtoggle	switchcmd, sid ain	Toggeln der Steckdose ein/aus	"0" oder "1" (Steckdose aus oder an)
getswitchstate	switchcmd, sid ain	Ermittelt Schaltzustand der Steckdose	"0" oder "1" (Steckdose aus oder an), "inval" wenn unbekannt
getswitchpresent	switchcmd, sid ain	Ermittelt Verbindungsstatus des Aktors	"0" oder "1" für Gerät nicht verbunden bzw. verbunden.  Bei Verbindungsverlust wechselt der Zustand erst mit einigen Minuten Verzögerung zu "0".
getswitchpower	switchcmd, sid ain	Ermittelt aktuell über die Steckdose entnommene Leistung	Leistung in mW, "inval" wenn unbekannt
getswitchenergy	switchcmd, sid ain	Liefert die über die Steckdose entnommene Energiemenge seit Erstinbetriebnahme oder Zurücksetzen der Energiestatistik	Energie in Wh, "inval" wenn unbekannt
getswitchname	switchcmd, sid ain	Liefert Bezeichner des Aktors	Name
getdevicelistinfos	switchcmd, sid	Liefert die grundlegenden Informationen aller SmartHome-Geräte	XML-Format mit grundlegenden und funktionspezifischen Informationen
gettemperature	switchcmd, sid ain	Letzte Temperaturinformation des Aktors	Temperatur-Wert in 0,1 °C, negative und positive Werte möglich Bsp. „200“ bedeutet 20°C
gethkrtsoll	switchcmd, sid ain	Für HKR aktuell eingestellte Solltemperatur	Temperatur-Wert in 0,5 °C, Wertebereich: 16 - 56 8 bis 28°C, 16 <= 8°C, 17 = 8,5°C..... 56 >= 28°C 254 = ON , 253 = OFF
gethkrkomfort	switchcmd, sid ain	Für HKR-Zeitschaltung eingestellte Komforttemperatur	Temperatur-Wert in 0,5 °C, Wertebereich: 16 - 56 8 bis 28°C, 16 <= 8°C, 17 = 8,5°C..... 56 >= 28°C 254 = ON , 253 = OFF
gethkrabsenk	switchcmd, sid ain	Für HKR-Zeitschaltung eingestellte Spartemperatur	Temperatur-Wert in 0,5 °C, Wertebereich: 16 - 56 8 bis 28°C, 16 <= 8°C, 17 = 8,5°C..... 56 >= 28°C 254 = ON , 253 = OFF
sethkrtsoll	switchcmd, sid ain param: HKR-Solltemperatur	HKR Solltemperatur einstellen. Mit dem „param“ Get-Parameter wird die Solltemperatur übergeben. Temperatur-Wert in 0,5 °C, Wertebereich: 16 - 56 8 bis 28°C, 16 <= 8°C, 17 = 8,5°C..... 56 >= 28°C 254 = ON , 253 = OFF	
getbasicdevicestats	switchcmd, sid ain	Liefert die grundlegenden Statistiken(Temperatur, Spannung, Leistung, Energie) des Aktors	XML-Format mit vorhandenen Statistiken

Kommando	HTTP-Parameter	Aktion	Antwort
gettemplatelistinfos	switchcmd, sid	Liefert die grundlegenden Informationen aller Vorlagen/Templates	XML-Format mit grundlegenden Informationen
applytemplate	switchcmd, sid ain	Vorlage anwenden, der ain-Parameter wird ausgewertet	
setsimpleonoff	switchcmd, sid, ain onoff:0(aus), 1(an) oder 2(toggle)	Gerät/Aktor/Lampe an-/ausschalten oder toggeln	
setlevel	switchcmd, sid, ain level:0(0%) bis 255(100%)	Dimm-, Höhen-, Helligkeit- bzw. Niveau-Level einstellen	
setlevelpercentage	switchcmd, sid, ain level:0(0%) bis 100(100%)	Dimm-, Höhen-, Helligkeit- bzw. Niveau-Level in Prozent einstellen	
setcolor	switchcmd, sid, ain hue: in Grad, 0 bis 359 (0° bis 359°), saturation:0(0%) bis 255(100%) duration: Schnelligkeit der Änderung in 100ms. 0 sofort	HueSaturation-Farbe einstellen Der HSV-Farbraum wird mit dem HueSaturation-Mode unterstützt. Der Hellwert(Value) kann über setlevel/setlevelpercentage konfiguriert werden, die Hue- und Saturation-Werte sind hier konfigurierbar.	
setcolortemperature	switchcmd, sid, ain temperature: in Kelvin, typisch im Bereich 2700 bis 6500 duration: Schnelligkeit der Änderung in 100ms. 0 sofort	Farbtemperatur einstellen	
getcolordefaults	switchcmd, sid	Liefert einen Vorschlag für die Werte der Color-/Farbauswahl	XML-Format mit <colordefaults>
sethkrboost	switchcmd, sid, endtimestamp, ain	HKR Boost aktivieren mit End-Zeit(Zeit in Sekunden seit 1970) zum Deaktivieren:endtimestamp=0 Die End-Zeit darf maximal bis zu 24 Stunden in der Zukunft liegen.	endtimestamp wenn erfolgreich
sethkrwindowopen	switchcmd, sid, endtimestamp, ain	HKR Fenster-offen Modus aktivieren mit End-Zeit(Zeit in Sekunden seit 1970) zum Deaktivieren:endtimestamp=0 Die End-Zeit darf maximal bis zu 24 Stunden in der Zukunft liegen.	endtimestamp wenn erfolgreich
setblind	switchcmd, sid, ain, target:: „open“, „close“ oder „stop“	Rollo schliessen(close), öffnen(open) oder stoppen(stop) Rollos haben den HANFUN-Unittyp Blind(281)	

Tabelle 2: Kommandos

### 3.1 Beispiele für switchcmd Aufrufe

<http://fritz.box/webservices/homeautoswitch.lua?switchcmd=setsimpleonoff&ain=13077%200012360-1&onoff=1>

<http://fritz.box /webservices/homeautoswitch.lua?switchcmd=setlevel&ain=13077%200012360-1&level=255>

<http://fritz.box /webservices/homeautoswitch.lua?switchcmd=setcolor&ain=13077%200012360-1&hue=100&saturation=255&duration=0>

fritz.box/webservices/homeautoswitch.lua?switchcmd=setcolortemperature&ain=13077 0000060-1&temperature=3000&duration=0

### 3.2 XML-Format von getdevicelistinfos Kommando

XML-Root-Knoten ist <devicelist> mit version-Attribut. Aktuelle Version ist 1.

Je bekanntem Gerät folgen <device> bzw. <group> Knoten. Im Aufbau sind bei <device> und <group> identisch, nur das <group> noch den <groupinfo> Knoten enthält.

Attribute von <device/group>:

- identifier: eindeutige ID, AIN, MAC-Adresse
- id: interne Geräteid
- fwversion: Firmwareversion des Gerätes
- manufacturer: "AVM"
- productname: Produktname des Gerätes, leer bei unbekanntem/undefiniertem Gerät
- functionbitmask: Bitmaske der Geräte-Funktionsklassen, beginnen mit Bit 0, es können mehrere Bits gesetzt sein
  - Bit 0: HAN-FUN Gerät
  - Bit 2: Licht/Lampe
  - Bit 4: Alarm-Sensor
  - Bit 5: AVM-Button
  - Bit 6: Heizkörperregler
  - Bit 7: Energie Messgerät
  - Bit 8: Temperatursensor
  - Bit 9: Schaltsteckdose
  - Bit 10: AVM DECT Repeater
  - Bit 11: Mikrofon
  - Bit 13: HAN-FUN-Unit
  - Bit 15: an-/ausschaltbares Gerät/Steckdose/Lampe/Aktor
  - Bit 16: Gerät mit einstellbarem Dimm-, Höhen- bzw. Niveau-Level
  - Bit 17: Lampe mit einstellbarer Farbe/Farbtemperatur

Beispiel FD300: binär 101000000(320), Bit6(HKR) und Bit8(Temperatursensor) sind gesetzt

Unterknoten von <device>/<group>

<present>0/1 - Gerät verbunden nein/ja

<txbusy>0/1 - das Senden eines Kommandos(wie Schaltbefehl oder Helligkeit ändern) läuft - ja(1) bzw. nein(0)

<name>Gerätename

optionale Unterknoten von <device>/<group> - wenn vom Gerät unterstützt

<batterylow>0 oder 1: Batterieladezustand niedrig - bitte Batterie wechseln

<battery>Batterieladezustand in Prozent

Es folgen Knoten für die verschiedenen Funktionsgruppen der Geräte (siehe functionbitmask). Nur grundsätzlich unterstützte Funktionsgruppen werden übermittelt.

Bspw. gibt es die <temperature>-Funktionsgruppe nur bei Geräten mit Temperatursensor.

Schaltsteckdose

<switch>

<state>0/1 - Schaltzustand aus/an (leer bei unbekannt oder Fehler)

<mode>"auto" oder "manuell" -> automatische Zeitschaltung oder manuell schalten (leer bei

unbekannt oder Fehler)

<lock>0/1 - Schaltsperre über UI/API ein nein/ja(leer bei unbekannt oder Fehler)

<devicelock>0/1 - Schaltsperre direkt am Gerät ein nein/ja(leer bei unbekannt oder Fehler)

Energie Messgerät

<powermeter>

<power>Wert in 0,001 W (aktuelle Leistung, wird etwa alle 2 Minuten aktualisiert)

<energy>Wert in 1.0 Wh (absoluter Verbrauch seit Inbetriebnahme)

<voltage>Wert in 0,001 V (aktuelle Spannung, wird etwa alle 2 Minuten aktualisiert)

Temperatursensor

<temperature>

<celsius>Wert in 0,1 °C, negative und positive Werte möglich

<offset>Wert in 0,1 °C, negative und positive Werte möglich

Alarmsensor

<alert>

<state>0/1 - letzter übermittelter Alarmzustand (leer bei unbekannt oder Fehler)

<lastalertchgtimestamp>Zeitpunkt der letzten Alarmzustandsänderung, timestamp in Sekunden seit 1970, 0 oder leer bei unbekannt

Taster

<button>

<lastpressedtimestamp>Zeitpunkt des letzten Tastendrucks, timestamp in Sekunden seit 1970, 0 oder leer bei unbekannt

optionale <button>-Attribute:

- identifier: eindeutige ID, AIN
- id: interne Geräteid

optionale <button> Unterknoten:

<name>Name

ACHTUNG: Ein <device> kann gegebenenfalls mehrere <button>-Knoten haben. Der FRITZ!DECT 400 hat 2 <button>-Knoten.

HAN-FUN Unit

Das HAN-FUN -Gerät taucht in der getdevicelistinfo Auflistung als HAN-FUN(ETSI)-Gerät und zusätzlich mit einem <device> je HAN-FUN-Unit auf. Die HAN-FUN Gerät ↔ Units Zuordnung erfolgt über die <etsiunitinfo> der HANFUND-Unit. Der Identifier einer HANFUND-Unit endet typischerweise auf "-" und <Nummer>.

Der FRITZ!DECT 500 ist ein HAN-FUN-Gerät mit einer Unit vom Typ „dimmbare Farb-Lampe“. Diese Unit kann die Farbe/Farbtemperatur ändern, die Helligkeit einstellen und an-/ausschalten.

<etsiunitinfo>

<etsideviceid> interne GeräteID des dezugehörigen HAN-FUN Gerätes

<unittyp> HAN-FUN Unit Typ

273 = SIMPLE\_BUTTON



256 = SIMPLE\_ON\_OFF\_SWITCHABLE  
 257 = SIMPLE\_ON\_OFF\_SWITCH  
 262 = AC\_OUTLET  
 263 = AC\_OUTLET\_SIMPLE\_POWER\_METERING  
 264 = SIMPLE\_LIGHT  
 265 = DIMMABLE\_LIGHT  
 266 = DIMMER\_SWITCH  
 277 = COLOR\_BULB  
 278 = DIMMABLE\_COLOR\_BULB  
 281 = BLIND  
 282 = LAMELLAR  
 512 = SIMPLE\_DETECTOR  
 513 = DOOR\_OPEN\_CLOSE\_DETECTOR  
 514 = WINDOW\_OPEN\_CLOSE\_DETECTOR  
 515 = MOTION\_DETECTOR  
 518 = FLOOD\_DETECTOR  
 519 = GLAS\_BREAK\_DETECTOR  
 520 = VIBRATION\_DETECTOR  
 640 = SIREN

#### <interfaces>HAN-FUN Interfaces

277 = KEEP\_ALIVE  
 256 = ALERT  
 512 = ON\_OFF  
 513 = LEVEL\_CTRL  
 514 = COLOR\_CTRL  
 772 = SIMPLE\_BUTTON  
 1024 = SUOTA-Update

an-/ausschaltbares Gerät/Steckdose/Lampe/Aktor

#### <simpleonoff>

<state> aktueller Schaltzustand, 0:aus, 1:an

Gerät mit einstellbarem Dimm-, Höhen-, Helligkeit- bzw. Niveau-Level

#### <levelcontrol>

<level> Level/Niveau von 0(0%) bis 255(100%)

<levelpercentage> Level/Niveau in Prozent, 0 bis 100 Prozent

Lampe mit einstellbarer Farbe/Farbtemperatur

#### <colorcontrol>

Attribute von <colorcontrol>

- supported\_modes: Bitmaske -- 0x01 = HueSaturation-Mode, 0x04 = Farbtemperatur-Mode
- current\_mode: string 1(HueSaturation), 4 (Farbtemperatur) oder ""(leer → unbekannt)

<hue/> Hue-Wert in Grad, 0 bis 359 (0° bis 359°), Achtung nur, wenn current\_mode == 1(HueSaturation) ansonsten leer/undefiniert

Der HSV-Farbraum wird mit dem HueSaturation-Mode unterstützt. Der Hellwert(Value) kann über setlevel/setlevelpercentage konfiguriert werden, die Hue- und Saturation-Werte sind über setcolor konfigurierbar.

<saturation/> Saturation-Wert von 0(0%) bis 255(100%), Achtung nur, wenn current\_mode == 1(HueSaturation) ansonsten leer/undefiniert

<temperature/> Wert in Kelvin, ein typischer Wertebereich geht von etwa 2700° bis 6500°

Heizkörperregler

<hkr>

<tist>Isttemperatur in 0,5 °C, Wertebereich: 0x0 - 0x64  
0 <= 0°C, 1 = 0,5°C..... 120 = 60°C, 254 = ON , 253 = OFF

<tsoll>Solltemperatur in 0,5 °C, Wertebereich: 0x10 - 0x38  
16 - 56 (8 bis 28°C), 16 <= 8°C, 17 = 8,5°C..... 56 >= 28°C, 254 = ON , 253 = OFF

<komfort>Komforttemperatur in 0,5 °C, Wertebereich: 0x10 - 0x38  
16 - 56 (8 bis 28°C), 16 <= 8°C, 17 = 8,5°C..... 56 >= 28°C, 254 = ON , 253 = OFF

<absenk>Absenkttemperatur in 0,5 °C, Wertebereich: 0x10 - 0x38  
16 - 56 (8 bis 28°C), 16 <= 8°C, 17 = 8,5°C..... 56 >= 28°C, 254 = ON , 253 = OFF

<batterylow>0 oder 1: Batterieladezustand niedrig - bitte Batterie wechseln

<battery>Batterieladezustand in Prozent

<>windowopenactiv> Fenster-offen Modus aktiviert: 0 oder 1

<>windowopenactiveendtime> Fenster-offen End-Zeit, in Sekunden seit 1970

<boostactive> Boost Modus aktiviert: 0 oder 1

<boostactiveendtime> Boost End-Zeit, in Sekunden seit 1970

<holidayactive> befindet sich der HKR aktuell in einem Urlaubszeitraum, 0 oder 1

<summeractive> befindet sich der HKR aktuell im „Heizung aus“ Zeitraum, 0 oder 1

<lock>0/1 - Tastensperre über UI/API ein nein/ja(leer bei unbekannt oder Fehler), Achtung die Tastensperre wird automatisch bei <summeractive>==1 oder <holidayactive>==1 aktiviert

<deviceunlock>0/1 - Tastensperre direkt am Gerät ein nein/ja(leer bei unbekannt oder Fehler)

<nextchange>nächste Temperaturänderung

<endperiod>timestamp in Sekunden seit 1970, 0 bei unbekannt

<tchange>Zieltemperatur, Wertebereich siehe tsoll(255/0xff ist unbekannt/undefiniert)

</nextchange>

<errorcode>Fehlercodes die der HKR liefert (bspw. wenn es bei der Installation des HKRs Problem gab):

0: kein Fehler

1: Keine Adaptierung möglich. Gerät korrekt am Heizkörper montiert?

2: Ventilhub zu kurz oder Batterieleistung zu schwach. Ventilstößel per Hand mehrmals öffnen und schließen oder neue Batterien einsetzen.

3: Keine Ventilbewegung möglich. Ventilstößel frei?

4: Die Installation wird gerade vorbereitet.

5: Der Heizkörperregler ist im Installationsmodus und kann auf das Heizungsventil montiert werden.

6: Der Heizkörperregler passt sich nun an den Hub des Heizungsventils an.

bei Gruppe

<groupinfo>

<masterdeviceid>interne id des Master/Chef-Schalters, 0 bei "keiner gesetzt"

<members>interne ids der Gruppenmitglieder, kommasepariert

Hinweis: bei Fehlern oder unbekannten Werten sind die betreffenden Elemente leer. Beispiel für unbekannte Temperatur: <celsius></celsius>

Beispiel für XML-Antwort

```
<devicelist version="1">
  <device identifier="08761 0000434" id="17" functionbitmask="896" fwversion="03.33"
  manufacturer="AVM" productname="FRITZ!DECT 200">
    <present>1</present>
    <name>Steckdose</name>
  <switch>
    <state>1</state> <mode>auto</mode>
    <lock>0</lock> <devicelock>0</devicelock>
  </switch>
    <powermeter>
      <power>0</power> <energy>707</energy> <voltage>230252</voltage>
    </powermeter>
    <temperature> <celsius>285</celsius> <offset>0</offset> </temperature>
  </device>
  <device identifier="08761 1048079" id="16" functionbitmask="1280" fwversion="03.33"
  manufacturer="AVM" productname="FRITZ!DECT Repeater 100">
    <present>1</present>
    <name>FRITZ!DECT Rep 100 #1</name>
    <temperature> <celsius>288</celsius> <offset>0</offset> </temperature>
  </device>
  <group identifier="65:3A:18-900" id="900" functionbitmask="512" fwversion="1.0"
  manufacturer="AVM" productname="">
    <present>1</present>
    <name>Gruppe</name>
    <switch> <state>1</state> <mode>auto</mode> <lock/> <devicelock/> </switch>
    <groupinfo> <masterdeviceid>0</masterdeviceid> <members>17</members> </groupinfo>
  </group>
</devicelist>
```

### 3.3 XML-Format von getbasicdevicestats Kommando

XML-Root-Knoten ist <devicestats> . Der ain-HTTP-Parameter identifiziert den Akteur.

Für Akteure mit Temperatursensor gibt es das <temperature> Element.

Für Akteure mit Energie Messgerät gibt es das <voltage>, <power> und <energy> Element.

Wenn eine Statistik vorhanden ist, dann gibt es im <temperature/voltage/power/energy>-Element ein oder mehrere <stats>-Elemente. Bei nicht verbundenen Akteuren ist kein <stats>-Element vorhanden.

Die Attribute von <stats> sind „count“ für Anzahl der Werte und „grid“ für den zeitliche Abstand/Auflösung in Sekunden. Der Inhalt von <stats> ist eine count-Anzahl kommasgetrennte Liste von Werten. Werte mit „-“ sind unbekannt/undefiniert.

Die Genauigkeit/Einheit der <temperature>-Werte ist 0,1°C.

Die Genauigkeit/Einheit der <voltage>-Werte ist 0,001V.

Die Genauigkeit/Einheit der <power>-Werte ist 0,01W.

Die Genauigkeit/Einheit der <energy>-Werte ist 1 Wh.

Beispiel für XML-Antwort

```
<devicestats>
```



Je Vorlage/Template folgen <template> Knoten.

Attribute von <template>:

- identifier: eindeutige string ID
- id: interne Template-ID
- functionbitmask: Bitmaske der Geräte-Funktionsklassen, beginnen mit Bit 0, es können mehrere Bits gesetzt sein

Die Unterknoten von <template>:

- <name> Template/Vorlagen Name
- <devices> <device>-List der zugehörigen Geräte  
Attribute von <device>: identifier: eindeutige string ID
- <applymask> - Unterknoten je nachdem welche Konfiguration gesetzt wird  
Unterknoten von <applymask>:  
    <hkr\_summer> //HKR Heizung-Aus-Schaltung (im Sommer)  
    <hkr\_temperature> //HKR Solltemperatur  
    <hkr\_holidays> //HKR Urlaubsschaltungen  
    <hkr\_time\_table> //HKR Zeitschaltung  
    <relay\_manual> //Relay AN/AUS  
    <relay\_automatic> //Relay Zeitschaltung

Beispiel für XML-Antwort

```
<devicestats>
<temperature>
<templatelist version="1">
<template identifier="tmp653A18-38AE7FDE9" id="60001" functionbitmask="320" applymask="10">
<name>Wohnzimmer Wochenende</name>
<devices>
  <device identifier="09995 0001012"/>
  <device identifier="09995 0000645"/>
</devices>
<applymask>
  <hkr_temperature/>
  <hkr_time_table/>
</applymask>
</templatelist>
```

### 3.5 XML-Format von getcolordefaults Kommando

Der Root-Knoten ist <colordefaults>, darunter gibt es den <hsdefaults>-Knoten für die HueSaturation-Werte und den <temperaturedefaults>-Knoten für die Farbtemperatur-Werte.

Der <hsdefaults>-Inhalt sind 12 <hs>-Elemente(mit hue\_index-Attribut). Dazu gehört 1 <name>-Unterelement(Farbname und eindeutiges Farb-Enum Attribut) und jeweils 3 <color>-Unterelemente(mit saturation\_index/sat\_index, hue. sat/saturation und val/value Attribut) Die hue-Wertebereich kann 0 bis 359 Grad sein. Der Saturation-Wertebereich geht von 0(0%) bis 255(100%). Der Value-Wertebereich geht ebenfalls von 0 (0%) bis 255 (100%), der Wert ist nur für die Anzeige der Farben interessant und wird beim Setzen nicht an die FRITZ!Box übermittelt.

Der <temperaturedefaults>-Inhalt sind 10 <temp>-Elemente mit value-Attribut. Der value-Attribut Wert ist die Farbtemperatur in Kelvin. Ein typischer Wertebereich geht von etwa 2700° bis 6500°.

Von den Colordefaults abweichende HueSaturation- oder Farbtemperatur-Werte sind unzulässig.  
Abweichende Werte werden vom setcolor-switchcmd bzw. setcolortemperatur-switchcmd verworfen.

Beispiel für XML-Antwort

```
<colordefaults>
<hsdefaults>
<hs hue_index="1">
<name enum="5569">Rot</name>
  <color sat_index="1" hue="358" sat="179" val="227"/>
  <color sat_index="2" hue="358" sat="112" val="237"/>
  <color sat_index="3" hue="358" sat="54" val="245"/>
</hs>
<hs hue_index="2">
<name enum="5570">Orange</name>
  <color sat_index="1" hue="35" sat="214" val="255"/>
  <color sat_index="2" hue="35" sat="140" val="255"/>
  <color sat_index="3" hue="35" sat="72" val="255"/>
</hs>
<hs hue_index="3"><name enum="5571">Gelb</name><color sat_index="1" hue="52" sat="153"
val="252"/><color sat_index="2" hue="52" sat="102" val="252"/><color sat_index="3" hue="52"
sat="51" val="255"/></hs>
<hs hue_index="4"><name enum="5572">Grasgrün</name><color sat_index="1" hue="92"
sat="123" val="248"/><color sat_index="2" hue="92" sat="79" val="250"/><color sat_index="3"
hue="92" sat="38" val="252"/></hs>
<hs hue_index="5"><name enum="5573">Grün</name><color sat_index="1" hue="107" sat="130"
val="220"/><color sat_index="2" hue="107" sat="82" val="232"/><color sat_index="3" hue="107"
sat="38" val="242"/></hs>
<hs hue_index="6"><name enum="5574">Türkis</name><color sat_index="1" hue="155" sat="133"
val="235"/><color sat_index="2" hue="155" sat="84" val="242"/><color sat_index="3" hue="155"
sat="41" val="248"/></hs>
<hs hue_index="7"><name enum="5575">Cyan</name><color sat_index="1" hue="191" sat="179"
val="255"/><color sat_index="2" hue="191" sat="118" val="255"/><color sat_index="3" hue="191"
sat="59" val="255"/></hs>
<hs hue_index="8"><name enum="5576">Himmelblau</name><color sat_index="1" hue="218"
sat="169" val="252"/><color sat_index="2" hue="218" sat="110" val="252"/><color sat_index="3"
hue="218" sat="56" val="255"/></hs>
<hs hue_index="9"><name enum="5577">Blau</name><color sat_index="1" hue="225" sat="204"
val="255"/><color sat_index="2" hue="225" sat="135" val="255"/><color sat_index="3" hue="225"
sat="67" val="255"/></hs>
<hs hue_index="10"><name enum="5578">Violett</name><color sat_index="1" hue="266" sat="169"
val="250"/><color sat_index="2" hue="266" sat="110" val="250"/><color sat_index="3" hue="266"
sat="54" val="252"/></hs>
<hs hue_index="11"><name enum="5579">Magenta</name><color sat_index="1" hue="296"
sat="140" val="250"/><color sat_index="2" hue="296" sat="92" val="252"/><color sat_index="3"
hue="296" sat="46" val="255"/></hs>
<hs hue_index="12"><name enum="5580">Pink</name><color sat_index="1" hue="335" sat="163"
val="242"/><color sat_index="2" hue="335" sat="107" val="248"/><color sat_index="3" hue="335"
sat="51" val="250"/></hs>
</hsdefaults>
<temperaturedefaults><temp value="2700" /><temp value="3000" /><temp value="3400" /><temp
value="3800" /><temp value="4200" /><temp value="4700" /><temp value="5300" /><temp
value="5900" /><temp value="6500" />
```

</temperaturedefaults>

</colordefaults>

## 4 Fehlercodes

Folgende HTTP Statuscodes werden verwendet:

Statuscode	Erläuterung
200 OK	Kommando wurde ausgeführt. Antwort entsprechend Tabelle 2: Kommandos, oder "inval", wenn kein Wert ermittelt werden konnte
400 Bad Request	HTTP Request fehlerhaft
403 Forbidden	Session-ID ungültig oder Benutzer nicht autorisiert
500 Internal Server Error	Interner Fehler

Tabelle 3: Fehlercodes

## Externe Referenzen

- [1] TR-064 First Steps, [https://avm.de/fileadmin/user\\_upload/Global/Service/Schnittstellen/AVM\\_TR-064\\_first\\_steps.pdf](https://avm.de/fileadmin/user_upload/Global/Service/Schnittstellen/AVM_TR-064_first_steps.pdf)
- [2] Session-IDs im FRITZ!Box Webinterface, [http://avm.de/fileadmin/user\\_upload/Global/Service/Schnittstellen/AVM\\_Technical\\_Note\\_-\\_Session\\_ID.pdf](http://avm.de/fileadmin/user_upload/Global/Service/Schnittstellen/AVM_Technical_Note_-_Session_ID.pdf)