AVM
Home Automation

HTTP Interface

05.08.25

FRITZ!

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	7.20	Color, Level und HANFUN-Aktualisierung Getdevicelistinfos-XML um 3 Unterkonten erweitert:‹simpleonoff›, ‹levelcontrol› und ‹colorcontrol› 4 neue switchcmd: setsimpleonoff, setlevel, setcolor, setcolortemperature
1.25	28.06.2019	7.20	getcolordefaults switchcmd hinzugefuegt
1.26	11.10.2019	7.20	HAN-FUN Einführung ‹txbusy› bei getdevicelistinfos XML hinzugefügt
1.27	25.11.2019	7.20	Hinweis zu unzulässigen Farb- und Farbtemperaturwerten minimal FRITZ!OS-Version aktualisiert
1.28	5.12.2019	7.20	‹levelpercentage›-Info und setlevelpercentage switchcmd hinzugefügt
1.29	14.01.2020	7.20	Hinweis zu HSV-Farbraum im HueSaturation-Mode
1.30	15.01.2020	7.20	Getcolordefaults liefert zusätzlich zu Hue uns Saturation auch Value
1.31	30.01.2020	7.20	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	7.20	‹windowopenactiv›
1.33	24.04.2020	7.20	neues switchcmd setblind
1.34	5.8.2020	7.20	Rollo Alarmbits, functionbitmask
1.35	12.8.2020	7.24	neues switchcmd: setname, startulesubscription und getsubscriptionstate
1.36	16.9.2020	7.24	HANFUN IF OPEN_CLOSE, OPEN_CLOSE_CONFIG Vorlagen Applymask erweitert(level, color, dialhelper) ‹humidity› für FD440, getbasicdevicestats mit ‹humidity› Statistik

Version	Datum	Ab FRITZ!OS	Änderung
			Switchcmd getdeviceinfos hinzugefügt Info zu Berechtigungen hinzugefügt
1.37	26.4.2021	7.24	setcolor und setcolortemperature duration-HTTP-Parameter wird aktuell nicht unterstützt
1.38	12.5.2021	7.24	neues switchcmd: setunmappedcolor Hinweis auf das Setzen von ColorDefaults abweichender HueSaturation-Werte
1.39	2.8.2021	7.39	Vorlage mit autocreate-Flag und metadata-String, neues cmd setmetadata für Vorlagen, ‹sub_templates›-Info für zugeordnete Unter-Vorlagen Statistik mit datatime-Unix-Timestamp Attribute
1.40	3.9.2021	7.39	Routinen/Trigger-Liste abfragen und aktivieren/deaktivieren neue cmds gettriggerlistinfos und settriggeractive gettriggerlistinfos mit Name, Identifier und active-Info(0/1)
1.41	3.9.2021	7.39	Lampen-Farb-Vorlage anlegen mit neuem addcolorleveltemplate cmd
1.42	19.10.2021	7.39	Vorlage applymask aktualisiert(timer_control, switch_master, http_request, tam_control, guest_wifi, main_wifi, sub_templates, sun_simulation)
1.43	19.10.2021	7.39	Vorlagen Beispiel mit sub_templates erweitert
1.44	18.11.2021	7.39	Neuer Vorlagen Typ: custom_notification
1.45	06.01.2022	7.39	Colorcontrol: fullcolorsupport, mapped, unmapped_hue und unmapped_saturation hinzugefügt ausserdem: rel_humidity, blind mit mode und endpositionsset
1.46	9.02.2022	7.39	windowopenactiveendtime bei externem Fenstersensor: -1
1.47	15.02.2022	7.39	adaptiveHeatingActive und adaptiveHeatingActive Info
1.48	22.02.2022	7.39	addcolorleveltemplate cmd Info erweitert
1.49	24.02.2022	7.39	Metadata-Info erweitert
1.50	23.03.2022	7.39	API-Mindestanforderung teils auf 7.39 korrigiert addcolorleveltemplate-cmd mit levelPercentage (bis 100) statt level(bis 255) Parameter
1.51	28.03.2022	7.39	windowopenactiveendtime Zusatzinfo Vorlagen und Szenarien Info
1.52	27.04.2022	7.24	Setname cmd unterstützt Vorlagen und Szenarien
1.53	02.05.2022	7.24	HKR ‹tist› und ‹tsoll› können leer sein, wenn die Temperatur unbekannt ist
1.54	10.05.2022	7.39	setname cmd unterstützt nicht vordefinierte Szenarien und Routinen Info zu ‹windowopenactiv› und ‹windowopenactiveendtime› erweitert
1.55	08.07.2022	7.24	Hinweis auf proprietäre 0xf7.. HAN-FUN Interfaces, Name mit Länge 40 Bytes
1.56	24.08.2022	7.39	Optionaler colorpreset Parameter für addcolorleveltemplate cmd
1.57	19.09.2022	7.39	gettemplatelistinfos cmd zusätzlich mit ‹triggers /› Info-Liste
1.58	11.01.2023		HAN-FUN und Zigbee Unit-Konzept und die Identifier
1.59	10.02.2023		functionbitmask bits konkretisiert
1.60	31.07.2023		level und levelpercentage für Rollo konkretisiert
1.61	18.09.2023		Einleitung und Geräte bzw. Unit Identifier(AIN) konkretisiert
1.62	14.11.2024		setsimpleonoff aktualisiert
1.63	21.10:2024		template aktualisiert

Inhaltsverzeichnis

1 Einführung

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

Vorraussetzung 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 FRITZ!Aktoren

Die FRITZ!-Aktoren werden über ihren Identifier(AIN) identifiziert.

Dieser muss bei jeder Zustandsabfrage bzw. Zustandsänderung für ein einzelnes Gerät übergeben werden.

1.2 HAN-FUN und Zigbee Unit-Konzept

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

Das FRITZ!Smart Gateway unterstützt das Zigbee Protokoll mit der Zigbee Cluster Library der Connectivity Standards Alliance. Im Falle von Zigbee-Geräten werden die dort verwendeten Endpoints mit ihren enthaltenen Clustern als HAN FUN-kompatible Units mit zugehörigen Interfaces abgebildet.

HAN-FUN und Zigbee benutzt ein Konzept von ein, oder mehreren Units je Gerät. So kann es zum Beispiel mehrere Alarm-Units oder OnOff-Units je Gerät geben.

Achtung: Diese Geräte haben immer mindestens unterschiedliche Identifier, einmal der Geräte-Identifier(in FRITZ!Box-UI zu finden) und ein Identifier je Unit. Der Unit-Identifier kann nicht im FRITZ!Box-UI ausgelesen werden, wird aber für jede Kommunikation mit der Unit benötigt. Siebe hierzu 1.2.1.

Eine Unit hat einen Unit-Type(z.B. Bewegungsmelder oder dimmbare Farb-Lampe) und eine Liste von unterstützten Interfaces(z.B. Alert, OnOff, Color oder Level/Helligkeit).

1.2.1 Identifier für Gerät und Unit

Zigbee Identifier Beispiel:

Gerät: Z0017880108AFB58C

Unit: Z0017880108AFB58C**0B**

Hinweis: der Unit-Identifier des Zigbee-Geräts ist 2 Zeichen länger, insgesamt immer 19 Zeichen

HANFUN Identifier Beispiel:

Gerät: 13077 0000258

Unit: 13077 0000258-**1**

Hinweis: der Unit-Identifier des HANFUN-Geräts endet immer auf "-‹Unit-Nummer›", also "-1" in dem Beispiel

Alle Identifier, egal ob für Gerät oder Unit, sind über die getdevicelistinfos Schnittstelle abfragbar.

Beispielinhalte für ein Teil der Zigbee-Gerät Informationen:

‹device identifier="Z0017880108AFB58C" id="20000" **functionbitmask="1"** fwversion="1.53.3_r27175" manufacturer="0x100b" productname="Zigbee"›

Hinweis: ein Zigbee-Gerät hat immer die functionbitmask="1"

Beispielinhalte für ein Teil der Zigbee-Unit Informationen:

‹device identifier="Z0017880108AFB58C**0B**" id="2003" functionbitmask="237572" fwversion="0.0" manufacturer="0x100b" productname=""›

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-Specifikation in [1].
Dem HTTP-Request sind dazu folgende Parameter URL-encoded zu übergeben:

Paramter	**Bedeutung**
ain	Identifikation des Aktors oder Templates, z. B. "012340000123" oder MAC Adresse für Netzwerkgeräte
switchcmd	Auszuführendes Kommado, 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], Die AVM Home Automation Session benötigt immer die Smart-Home-Berechtigung. Ausserdem wird für einige Kommandos die "Eingeschränkte FRITZ!Box Einstellungen für Apps"-Berechtigung benötigt. Zu Berechtigungen siehe "Actions and User Rights" in der TR064-Specifikation in [1].

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, getsubscriptionstate und getbasicdevicestats: Die HTTP Response enthält den Inhalt 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 ain	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 Ernergiemenge 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 funktionsspezifischen 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
gettriggerlistinfos	switchcmd, sid	Liefert die grundlegenden Informationen aller Routinen/Trigger	XML-Format mit grundlegenden Informationen

Kommando	HTTP-Parameter	Aktion	Antwort
settriggeractive	switchcmd, sid, ain, active	Trigger aktivieren(active=1) oder deaktivieren(active=0)	
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) oder 1(an)	Gerät/Aktor/Lampe an-/ausschalten	
setlevel	switchcmd, sid, ain level:0(0%) bis 255(100%)	Dimm-, Rollo-, Höhen-, Helligkeit- bzw. Niveau-Level einstellen	
setlevelpercentage	switchcmd, sid, ain level:0(0%) bis 100(100%)	Dimm-, Rollo-, 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 Hinweis: duration wird aktuell nicht unterstützt	HueSaturation-Farbe einstellen, beschränkt auf die Colordefaults Farbauswahl 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.	
setunmappedcolor	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 Hinweis: duration wird aktuell nicht unterstützt	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 Hinweis: duration wird aktuell nicht unterstützt	Farbtemperatur einstellen	
addcolorleveltemplate	switchcmd, sid, name, levelPercentage hue und saturation ODER temperature ain-Geräteliste in Anzahl n child_‹n›-Parametern beginnend mit child_1, child_2.. optional: colorpreset	Erzeugt eine Farb-Vorlage für Lampen. Es muss name, levelPercentage, child_n und hue+saturation oder temperature übergeben werden. Wertebereiche wie bei setunmappedcolor, setcolortemperature und setlevelpercentage wenn colorpreset==“true“ dann werden die Colordefaults benutzt, ansonsten und im default(false) werden die Colordefaults nicht benutzt	
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	

Kommando	HTTP-Parameter	Aktion	Antwort
		Blind(281)	
setname	switchcmd, sid, ain, name	Gerät-, Gruppen-, Vorlagen- oder Szenarienname ändern. Routinen und vordefinierte Szenarien können nicht umbenannt werden. name int UTF8, maximal 40 Bytes der ain-Parameter entspricht dem identifier Achtung: die Benutzer-Session muss das Smarthome- UND App-Recht haben Hinweis: benötigt die "Eingeschränkte FRITZ!Box Einstellungen für Apps"-Berechtigung	
setmetadata	switchcmd, sid, ain, metadata	json-metadaten der Vorlage oder Leerstring ändern/setzen Maximalgrösse 200 byte, 2 definierte json-keys: "icon" – Integer, ID des Icons zum passenden Icon-Fonts "type" – Nur für Szenarien definiert: Typ ders Szenarios. Ein String aus "coming", "leaving" oder "generic" (generic für freie Szenarien). Einer der beiden keys muss gesetzt sein, kein anderer key ist zugelassen	
startulesubscription	switchcmd, sid	DECT-ULE-Geräteanmeldung starten Achtung: die Benutzer-Session muss das Smarthome- UND App-Recht haben Hinweis: benötigt die "Eingeschränkte FRITZ!Box Einstellungen für Apps"-Berechtigung	
getsubscriptionstate	switchcmd, sid	DECT-ULE-Geräteanmeldestatus abfragen Achtung: die Benutzer-Session muss das Smarthome- UND App-Recht haben Hinweis: benötigt die "Eingeschränkte FRITZ!Box Einstellungen für Apps"-Berechtigung	XML: ‹state›-Rootknoten mit code-Attribute: "0"=Anmeldung läuft nicht "1"=Anmeldung läuft "2"=timeout "3"=sonstiger Error Unterknoten ‹latestain/› enthält gegebenenfalls die AIN des zuletzt angemeldeten Geräts Beispiel: ‹state code="0"› ‹latestain›13077 0000258‹/latestain› ‹/state›
getdeviceinfos	switchcmd, sid, ain	Liefert die grundlegenden Informationen des SmartHome-Geräts	XML-Format mit grundlegenden und funktionsspezifischen Informationen siehe getdevicelistinfos

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

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

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.

Beim getdeviceinfos switchcmd wird der ‹device› bzw. ‹group› Knoten zurückgegeben.

Attribute von ‹device/group›:

- identifier: eindeutige ID, AIN, MAC-Adresse
  Achtung: für HANFUN- oder Zigbee-Geräte siehe auch Kapitel 1.1 zum Identifier bei Units
- 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: FRITZ! Button
  Bit 6: FRITZ! Heizkörperregler
  Bit 7: FRITZ! Energie Messgerät
  Bit 8: Temperatursensor
  Bit 9: FRITZ! Schaltsteckdose
  Bit 10: FRITZ! DECT Repeater
  Bit 11: FRITZ! 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
  Bit 18: Rollladen(Blind) - hoch, runter, stop und level 0% bis 100 %
  Bit 20: Luftfeuchtigkeitssensor

  Die Bits 5,6,7,9,10 und 11 werden nur von FRITZ!-Geräten verwendet und nicht von HANFUN- oder Zigbee-Geräten.

  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 folgenden 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›
Achtung: für HANFUN- oder Zigbee-Geräte siehe auch Kapitel 1.1 zum Identifier bei Units

‹state›0/1 - letzter übermittelter Alarmzustand (leer bei unbekannt oder Fehler)

Beim Rollladen als Bitmaske auszuwerten.
0000 0000 - Es liegt kein Fehler vor.
0000 0001 - Hindernisalarm, der Rollladen wird gestoppt und ein kleines Stück in entgegengesetzte Richtung bewegt.
0000 0010 - Temperaturalarm, Motor überhitzt.

‹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

FD440 Taster mit Luftfeuchtigkeitssensor

‹avmbutton›

‹humidity›‹rel_humidity›relative Luftfeuchtigkeit in Prozent von 0 bis 100, Spezialwert: -9999 bei unbekannt

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 HANFUN-Unit. Der Identifier einer HANFUN-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

‹unittype› 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
516 = OPEN_CLOSE
517 = OPEN_CLOSE_CONFIG
772 = SIMPLE_BUTTON
1024 = SUOTA-Update

an-/ausschaltbares Gerät/Steckdose/Lampe/Aktor

‹simpleonoff›
Achtung: für HANFUN- oder Zigbee-Geräte siehe auch Kapitel 1.1 zum Identifier bei Units

‹state› aktueller Schaltzutand, 0:aus, 1:an

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

‹levelcontrol›
Achtung: für HANFUN- oder Zigbee-Geräte siehe auch Kapitel 1.1 zum Identifier bei Units

‹level› Level/Niveau von 0(0%) bis 255(100%)

‹levelpercentage› Level/Niveau in Prozent, 0 bis 100 Prozent

Lampe mit einstellbarer Farbe/Farbtemperatur

‹colorcontrol›
Achtung: für HANFUN- oder Zigbee-Geräte siehe auch Kapitel 1.1 zum Identifier bei Units

Attribute von ‹colorcontrol›

- supported_modes: Bitmaske -- 0x01 = HueSaturation-Mode, 0x04 = Farbtemperatur-Mode
- current_mode: string 1(HueSaturation), 4 (Farbtemperatur) oder ""(leer ⇢ unbekannt)
- fullcolorsupport: bool, 0 oder 1, Lampe unterstützt setunmappedcolor, also von den colordefaults abweichende HueSaturation-Werte

- mapped: bool, 0 oder 1, 0: von den Colordefaults abweichend zugeodneter HueSaturation-Wert gesetzt, 1: Colordefaults Wert gesetzt

‹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 und setunmappedcolor konfigurierbar.

‹saturation/› Saturation-Wert von 0(0%) bis 255(100%), Achtung nur, wenn current_mode == 1(HueSaturation) ansonsten leer/undefiniert

‹unmapped_hue› wie ‹hue›, nur bei mapped=1 der über die Colordefaults korrigiert zugeordnete Hue-Wert

‹unmapped_saturation› wie ‹saturation›, nur bei mapped=1 der über die Colordefaults korrigiert zugeordnete Saturation-Wert

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

Rollo

‹blind›
Achtung: für HANFUN- oder Zigbee-Geräte siehe auch Kapitel 1.1 zum Identifier bei Units

Für die Rollo Höhenlevel Einstellung siehe ‹level›

‹mode› "auto" oder "manuell" -› automatische Zeitschaltung oder manuell fahren (leer bei unbekannt oder Fehler)

‹endpositionsset› ist die Endlage für das Rollo konfiguriert?

leer: unbekannt, 0: nicht konifiguriert, 1: konfiguriert

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
leer wenn die Isttemperatur unbekannt ist(bspw. wenn der HKR inaktiv ist, also present=0)

‹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
leer wenn die Solltemperatur unbekannt ist(bspw. wenn der HKR inaktiv ist, also present=0)

‹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›Absenktemperatur 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
Der Fenster-offen Modus kann entweder durch einen Temperaturabfall vom HKR selbst erkannt,
durch einen externen Tür-/Fenstersensor ausgelöst oder über die API mit sethkrwindowopen aktiviert worden sein.

‹windowopenactiveendtime› Fenster-offen End-Zeit, in Sekunden seit 1970
wenn ein externer Tür-/Fenstersensor zugeordnet und aktiv ist: windowopenactiveendtime = -1
in diesem Fall ist die End-Zeit unbekannt
Wenn der Fenster-offen Modus vom HKR selbst durch einen Temperaturabfall erkannt worden ist, wird die End-Zeit durch den in der FRITZ!OS Benutzeroberfläche konfigurierten Wert bestimmt.

Wenn der Fenster-offen Modus über sethkrwindowopen aktiviert wurde, wird die End-Zeit durch den entsprechenden Parameter dieses Kommandos bestimmt.

‹boostactive› Boost Modus aktiviert: 0 oder 1

‹boostactiveendtime› Boost End-Zeit, in Sekunden seit 1970

‹adaptiveHeatingActive› adaptive Heizregelung aktiviert, 0 oder 1

‹adaptiveHeatingRunning› 0 oder 1, heizt die adaptive Heizregelung aktuell
gegebenenfalls in der Zeit vor einem Komforttemperatur-Schaltpunkt aktiv

‹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

‹devicelock›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 Aktor.

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

Für Aktoren mit Luftfeuchtigkeitssensor gibt es das <humidity> Element.

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

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

Die Attribute von <stats> sind „count“ für Anzahl der Werte und „grid“ für den zeitliche Abstand/Auflösung in Sekunden. Das „datatime“ Attribute enthält den Unix-Timestamp der letzten Aktualisierung. Der Inhalt von <stats> ist eine count-Anzahl kommaseparierte 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.

Die Genauigkeit/Einheit der <humidity>-Werte ist Prozent.

Beispiel für XML-Antwort

<devicestats>

<temperature>

<stats count="96" grid="900" datatime="1627466380">

245,255,-,
-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-

</stats>

</temperature>

<voltage>

<stats count="360" grid="10" datatime="1627466380">

228615,228615,228615,228615,228615,228615,228615,228615,228615,228615,228615,228615,229040,229040,22
9040,229040,229040,229040,229040,229040,229040,229040,229040,229040,229882,229882,229882,229882,2298
82,229882,229882,229882,229882,229882,229882,229882,229796,229796,229796,229796,229796,229796,229796
,229796,229796,229796,229796,229796,228864,228864,228864,228864,228864,228864,228864,228864,228864,2
28864,228864,228864,229059,229507,228667,228507,228919,229180,229245,229067,229245,228832,229352,229
041,229041,229041,229041,229041,229041,229041,229041,226194,226194,226194,226194,226194,226194,22619
4,226194,226194,226194,226194,226194,226451,226451,226451,226451,226451,226451,226451,226451,226451,
226451,226451,226451,226894,226894,226894,226894,226894,226894,226894,226894,226894,226894,226894,22
6894,226894,227326,227541,227843,228241,227120,227171,227265,227836,227406,227598,228280,228152,2276
26,227820,227364,226775,227162,227003,228258,227902,228142,227950,227103,227003,227360,227456,227957
,227929,227274,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

</stats>

</voltage>

<power>

<stats count="360" grid="10" datatime="1627466380">

1067,1067,1067,1067,1067,1067,1067,1067,1067,1067,1067,1067,1066,1066,1066,1066,1066,1066,1066,1066,1066,1066,1066,1066,1082,1082,1082,1082,1082,1082,1082,1082,1082,1082,1082,1082,1086,1086,1086,1086,1086,1086,1086,1086,1086,1086,1086,1086,1085,1085,1085,1085,1085,1085,1085,1085,1085,1085,1085,1085,1097,1096,1093,1095,1096,1099,1100,1100,1098,1100,1102,1102,1102,1102,1102,1102,1102,1102,1102,1095,1095,1095,1095,1095,1095,1095,1095,1095,1095,1095,1106,1106,1106,1106,1106,1106,1106,1106,1106,1106,1106,1106,1129,1129,1129,1129,1129,1129,1129,1129,1129,1129,1129,1129,1129,1126,1129,1134,1132,1127,1129,1133,1138,1137,1141,1144,1146,1146,1153,1157,1165,1174,1195,1219,1246,1287,1329,1354,1341,1285,588,91,132,133,0

</stats>

</power>

<energy>

<stats count="12" grid="2678400" datatime="1627466324">

413,0,0,0,0,0,0,0,0,0,0,0

</stats>

<stats count="31" grid="86400" datatime="1627466324">

41,0,0,0,0,0,0,0,126,132,52,0,0,0,0,61,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

</stats>

</energy>

</devicestats>

3.4 XML-Format von gettemplatelistinfos Kommando

Vorlagen machen Einstellungen wiederverwendbar für mehrere Geräte und/oder Gruppen. Eine Vorlage kann sich immer nur auf einen Gerätetyp beziehen.

Szenarien machen Einstellungen basierend auf Vorlagen wiederverwendbar. Ein Szenario kann Vorlagen für mehrere Gerätetypen enthalten.

Es gibt vordefinierte und individuelle(freie) Szenarien. Bei Szenarien ist die applymask <sub_templates/> gesetzt.

Die automatisch erzeugten vordefinierten Szenarien werden aus automatisch erzeugten Vorlagen erstellt. Bei automatisch erzeugten Vorlagen und Szenarien ist das autocreate=1 Attribute gesetzt. Die automatisch erzeugten Vorlagen(autocreate=1, kein <sub_templates/> gesetzt) sollten im GUI ausgeblendet werden, da sie zum vordefinierten Szenario gehören und einen nicht änderbaren Defaultnamen bekommen.

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

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
- autocreate: 0/1-Flag wurde Vorlage automatisch erzeugt

Die Unterknoten von <template>:

- <name> Template/Vorlagen Name
- <metadata> json-metadaten oder Leerstring, Maximalgrösse 200 byte, 2 definierte json-keys:
  "icon" – Integer, ID des Icons zum passenden Icon-Fonts
  "type" – Nur für Szenarien definiert: Typ ders Szenarios. Ein String aus "coming", "leaving" oder "generic" (generic

für freie Szenarien).
Einer der beiden keys muss gesetzt sein, kein anderer key ist zugelassen

- ‹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› //an-/ausschaltbares Steckdose/Lampe/Aktor AN/AUS
  ‹relay_automatic› //an-/ausschaltbares Steckdose/Lampe/Rollladen Zeitschaltung
  ‹level› //Level bzw. Helligkeit von Lampe/Rollladen
  ‹color› //Farbe oder Farbtemperatur
  ‹dialhelper› //Rufansage
  ‹sun_simulation› //Licht Sonnenauf- und Sonnenuntergangsimulation
  ‹sub_templates› //gruppierte Templates, Szenarien
  ‹main_wifi› //WLAN an/aus
  ‹guest_wifi› //*Gast-WLAN an/*aus
  ‹tam_control› //Anrufbeantworter an/aus
  ‹http_request› //beliebigen HTTP-Request versenden
  ‹timer_control› //HKR Boost/Fenster auf/Temperatur-Override aktivieren
  ‹switch_master› //Geräte auf Zustand anderer Geräte schalten
  ‹custom_notification› //Pushmail/App-Notification auslösen
- ‹sub_templates› der Vorlage zugeordete Unter-Vorlagen, je ‹template› mit identifier-Attribute
- ‹triggers› der Vorlage zugeordnete Routinen/Trigger, je ‹trigger› mit identifier-Attribute

Beispiel für XML-Antwort

‹templatelist version="1"›

‹template identifier="tmp653A18-38AE7FDE9" id="60001" functionbitmask="320" applymask="10" autocreate="0"›

‹name›Wohnzimmer Wochenende‹/name›

‹metadata /›

‹devices›

‹device identifier="09995 0001012"/›

‹device identifier="09995 0000645"/›

‹/devices›

‹sub_templates/›

‹applymask›

‹hkr_temperature/›

‹hkr_time_table/›

‹/applymask›

‹/template›

‹template identifier="tmpD0EF2E-3CE52B246" id="60100" functionbitmask="16384" autocreate="0" applymask="2048"›

‹name›Vorlagen Gruppe‹/name›

‹metadata/›

‹devices/›

‹sub_templates›

‹template identifier="tmp653A18-38AE7FDE9"/›

‹template identifier="tmpD0EF2E-4711"/›

‹/sub_templates›

‹applymask›

‹sub_templates/›

‹triggers/›

‹/applymask›

‹/template›

‹/templatelist›

3.5 XML-Format von gettriggerlistinfos Kommando

XML-Root-Knoten ist ‹triggerlist› mit version-Attribut. Aktuelle Version ist 1.

Je Routine/Trigger folgen ‹trigger› Knoten.

Attribute von ‹trigger›:

- identifier: eindeutige string ID
- active: 0/1-Flag, Trigger aktiviert(1) oder deaktiviert(0)

Die Unterknoten von ‹trigger›:

- ‹name› Routine/Trigger Name

Beispiel für XML-Antwort

‹triggerlist version="1"›

‹trigger identifier="trg695F2D-3CBF1DC25" active="1"›

‹name›Trigger AlertOn‹/name›

‹/trigger›

‹/triggerlist›

3.6 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 is 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 unzulässsig. Abweichende Werte werden vom setcolor-switchcmd bzw. setcolortemperatur-switchcmd verworfen.
Mit setunmappedcolor können von den Colordefaults abweichende HueSaturation-Werte gesetzt werden.

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	Erläuerung
200 OK	Kommando wurde ausgeführt. Antwort entsprechend Tabelle 2: Kommandos, oder "inval", wenn kein Wert ermittelt werden konnte
400 Bad Request	HTTP Request fehlerhaft, Parameter sind ungültig, nicht vorhanden oder der Wertebereich wurde überschritten
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://fritz.com/fileadmin/user_upload/Global/Service/Schnittstellen/AVM_TR-064_first_steps.pdf

[2] Session-IDs im FRITZ!Box Webinterface, http://fritz.com/fileadmin/user_upload/Global/Service/Schnittstellen/AVM_Technical_Note_-_Session_ID.pdf