



# HTTP / Web API

Technisches Handbuch

# Inhaltsverzeichnis

1	Sicherheit und Garantie.....	3
2	Allgemeines.....	5
2.1	Aufbau.....	5
3	Konfiguration.....	6
3.1	Erklärung Variablen.....	8
3.2	GET-Output.....	9
3.3	POST-Output.....	11
3.4	PUT-Output.....	12
3.5	GET-Input.....	13
4	IO-Konfiguration Beispiel.....	15
4.1	Licht.....	15

# 1 Sicherheit und Garantie

Die Geräte sind nach den derzeit gültigen Regeln der Technik gebaut und betriebssicher. Sie wurden geprüft und haben das Werk in sicherheitstechnisch einwandfreiem Zustand verlassen.

Dennoch gibt es Restgefahren. Lesen und beachten Sie die Sicherheitshinweise, um Gefahren zu vermeiden.

Für Schäden durch Nichtbeachtung von Sicherheitshinweisen übernimmt die Ekon GmbH keine Haftung.

## Verwendete Sicherheitshinweise

Die folgenden Hinweise weisen Sie auf besondere Gefahren im Umgang mit den Geräten hin oder geben nützliche Hinweise:

Hinweise in diesen Boxen sind generelle Tipps zum Text, die etwas hervorgehoben werden.



### HINWEIS

Das Signalwort Hinweis kennzeichnet nützliche Tipps und Empfehlungen für den effizienten Umgang mit dem Produkt.



### VORSICHT

#### Gesundheitliche Schäden / Sachschäden

Das jeweilige Warnsymbol in Verbindung mit dem Signalwort Vorsicht kennzeichnet eine Gefahr, die zu leichten (reversiblen) Verletzungen oder Sachschäden führen kann.



### WARNUNG

#### Schwere gesundheitliche Schäden

Das jeweilige Warnsymbol in Verbindung mit dem Signalwort Warnung kennzeichnet eine drohende Gefahr, die zum Tod oder zu schweren (irreversiblen) Verletzungen führen kann.



### GEFAHR

#### Lebensgefahr / Schwere gesundheitliche Schäden

Das jeweilige Warnsymbol in Verbindung mit dem Signalwort Gefahr kennzeichnet eine unmittelbar drohende Gefahr, die zum Tod oder zu schweren (irreversiblen) Verletzungen führt.

## Sicherheitshinweise

Im Folgenden sind die Sicherheitshinweise zum in diesem Dokument beschriebenen Produkt aufgelistet. Bitte beachten Sie diese bei der Verwendung des Produkts.



## GEFAHR

### Elektrische Spannung !

Lebensgefahr und Brandgefahr durch elektrische Spannung

Im Innern des Geräts befinden sich ungeschützte spannungsführende Bauteile. Die VDE-Bestimmungen beachten. Alle zu montierenden Leitungen spannungslos schalten und Sicherheitsvorkehrungen gegen unbeabsichtigtes Einschalten treffen. Das Gerät bei Beschädigung nicht in Betrieb nehmen. Das Gerät bzw. die Anlage außer Betrieb nehmen und gegen unbeabsichtigten Betrieb sichern, wenn anzunehmen ist, dass ein gefahrloser Betrieb nicht mehr gewährleistet ist.



## VORSICHT

### Geräteschaden durch äußere Einflüsse !

Feuchtigkeit und eine Verschmutzung der Geräte können zur Zerstörung der Geräte führen.

Schützen Sie die Geräte bei Transport, Lagerung und im Betrieb vor Feuchtigkeit, Schmutz und Beschädigungen.

## Garantie

Das Gerät ist ausschließlich für den sachgemäßen Gebrauch bestimmt. Bei jeder unsachgemäßen Änderung oder Nichtbeachten der Bedienungsanleitung erlischt jeglicher Gewährleistungs- oder Garantieanspruch. Nach dem Auspacken ist das Gerät unverzüglich auf mechanische Beschädigungen zu untersuchen. Wenn ein Transportschaden vorliegt, ist unverzüglich der Lieferant davon in Kenntnis zu setzen. Das Gerät darf nur als ortsfeste Installation betrieben werden, das heißt nur in montiertem Zustand und nach Abschluss aller Installations- und Inbetriebnahmearbeiten und nur im dafür vorgesehenen Umfeld. Für Änderungen der Normen und Standards nach Erscheinen der Bedienungsanleitung ist Ekon GmbH nicht haftbar.

## 2 Allgemeines

Mit dem myGEKKO können über das Netzwerk frei definierbare HTTP/S Befehle gesendet und empfangen werden.

So können Geräte, welche über HTTP/S Befehle verfügen, über myGEKKO ausgelesen oder Werte geschrieben werden. Die Befehle können in der IO Konfiguration definiert und in den jeweiligen Bausteinen verwendet werden.



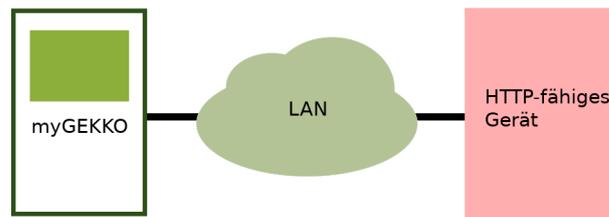
### HINWEIS

Die HTTP/S WebAPI eignet sich auch für das Steuern von Geräten über Cloud Diensten wie z. B. IFTTT (siehe dazu eigenes Tutorial).

### 2.1 Aufbau

Voraussetzung für die Verwendung der HTTPS Befehle ist, dass das Netzwerk korrekt konfiguriert ist und die Geräte untereinander kommunizieren können.

Kontrollieren Sie hierzu gegebenenfalls die Netzwerkeinstellungen.

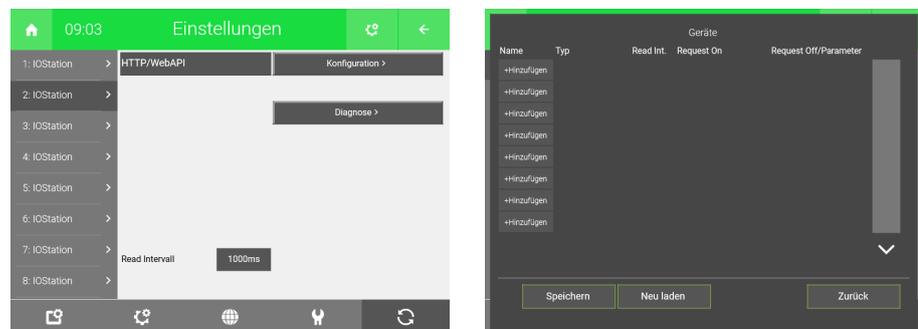


## 3 Konfiguration

Melden Sie sich als Konfigurator am myGEKKO an:

1. Gehen Sie in die Einstellungen (Zahnrad → "Einstellungen") und wechseln in die IO-Station Konfiguration (drehende Pfeile bzw. letzter Tab)
2. Legen Sie eine neue IO-Station an mit dem Stationstyp HTTP/WebAPI
3. Über „Konfiguration >>“ können Sie die verschiedenen HTTP Befehle definieren. Es sind 4 verschiedene HTTP Typen verfügbar:
  - siehe Kapitel 3.2
  - siehe Kapitel 3.3
  - siehe Kapitel 3.4
  - siehe Kapitel 3.5
4. Nach einem Klick auf "Diagnose >>" wird das Diagnosefenster angezeigt, welches Informationen und Statuscodes anzeigt
5. Im Feld neben "Read Intervall" kann der übergeordnete Leseintervall vergeben werden. Dieser kann zwischen 1 Sekunde und 65 Sekunden eingestellt werden

**Bsp.:** Wenn das übergeordnete Leseintervall auf 1 Sek definiert wurde, werden die Requests standardmäßig jede Sekunde ausgelesen. Wenn aber bei dem jeweiligen Request ein längerer Intervall definiert ist, wird dieser Request nur alle 5 Sekunden ausgelesen wie in dem nachfolgenden Beispiel der Dämmerung. Wurde der übergeordnete Leseintervall auf 10 Sek festgelegt, wird die Dämmerung auch nur jede 10 Sekunden ausgelesen.



In der Konfiguration können Sie die Befehle definieren:

1. Durch Tippen auf das Feld "+ Hinzufügen" können Sie einen Namen für den http-Befehl definiert werden.

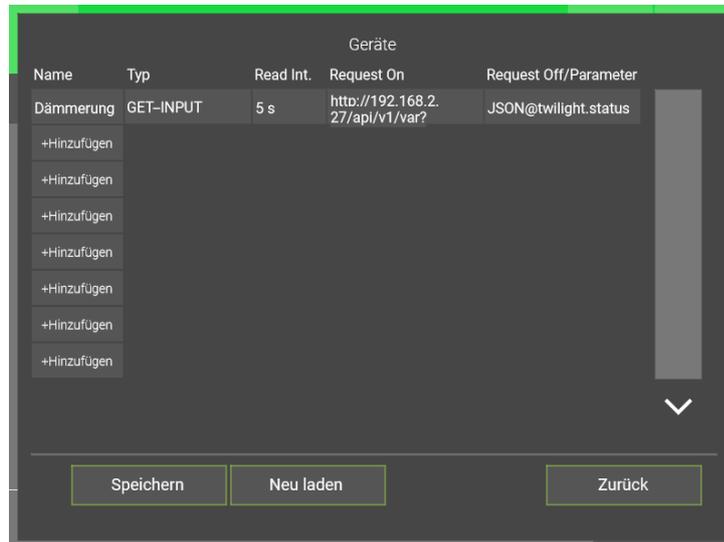


### HINWEIS

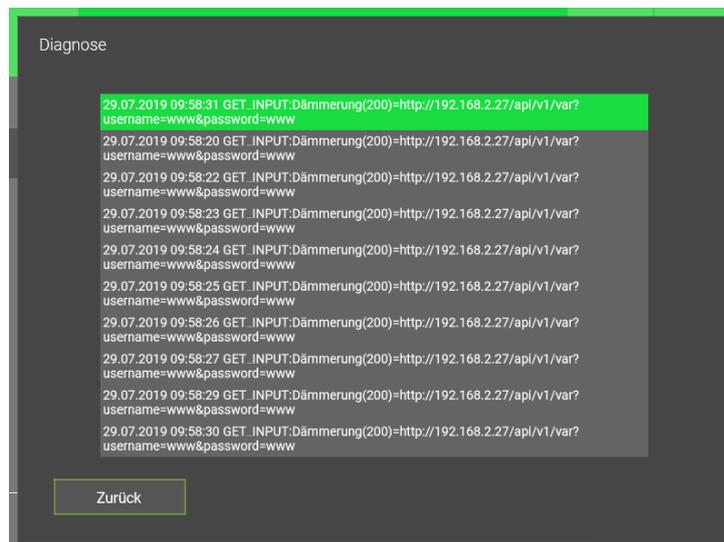
Nach der Eingabe des Namens sollte der Request-Befehl innerhalb 100 Sekunden eingegeben werden. Hierbei empfehlen wir die Eingabe des Requests über den myGEKKO Viewer, da die Tastenkürzel für Kopieren und Einfügen verwendet werden können.

2. Wählen Sie den Typ des HTTP-Befehl aus

3. Bei "Read intervall" kann der Leseintervall je Request definiert werden bei Typ GET-INPUT
4. Definieren Sie den Request On (Weitere Informationen zu den Requests finden Sie in den folgenden Kapiteln)
5. Definieren Sie den Request Off/Parameter
6. Speichern Sie Ihre Einstellungen und Befehle durch einen Klick auf "Speichern"



Im Diagnosefenster werden Informationen zu den Requests mit dem entsprechenden Statuscode angezeigt. Statuscode 200 bedeutet, dass der Request erfolgreich bearbeitet wurde und das Ergebnis der Anfrage in der Antwort übertragen wird. Eine detaillierte Beschreibung der HTTP Statuscodes finden Sie im Internet. ([Liste der HTTP Status Codes - Wikipedia](#))



## 3.1 Erklärung Variablen

**@VALUE@** Der Platzhalter @VALUE@ wird verwendet um einen Wert aus dem Element weiterzugeben:

- **AO:**  
Hier wird die Variable durch eine Ganze Zahl ersetzt (0,1,2,3,4, ...)
- **DO:**  
Hier wird die Variable durch 0 oder 1 ersetzt (bei >1 wird eine 1 ausgegeben)

**@VALUE:0[false],1[true]@** Dieser Platzhalter wird verwendet, um die Werte "true" oder "false" anstatt den standardmäßigen "0" und "1" zu verschicken:

- **Wörtern:**  
@VALUE:0[false],1[true]@ Hier wird die Variable mit dem übergebenen Wert als AO oder DO als Text ersetzt.  
Bsp.: Bei einer 0 wird false eingesetzt  
Bei einer 1 wird true eingesetzt

**TEXT@Parameter=@VALUE@** Diese Form wird vor Allem für das Einlesen von Werten als AI (analoger Input) verwendet. Der Platzhalter @VALUE@ ersetzt die Zahl nach dem Parameter/Text:

Bsp.: Hka\_Mw1.sWirkleistung=5,67  
TEXT@Hka\_Mw1.sWirkleistung=@VALUE@

**JSON@VALUE** Dieser Parameter wird zum Suchen eines Wertes in einem JSON-Objekt verwendet. Der Platzhalter VALUE wird mit dem gewünschtem Suchobjekt ersetzt. Für eine detaillierte Beschreibung der auslesbaren JSON und die dazu benötigten Query API des Controllers konsultieren Sie das Handbuch zu "my-GEKKO Query API".

Bsp.: JSON@twilight.status  
Der gefundene Wert wird umgewandelt, sodass dieser als AI verwendet werden kann.

Bsp.: JSON@item0.sumstate.value[0]  
JSON@temperatures[0].value  
JSON@lights.sumstate[3].value[0]  
Arrays (Felder an Werten) können mit eckigen Klammern [ ] abgerufen werden. Beachten Sie dass Array-Nummerierungen bei 0 starten. Arrays können auch ineinander verschaltet sein.

**JSON@FIELD@VALUE** Diese Form dient dazu im JSON Objekt nach dem Feld "FIELD" zu suchen und dessen Wert mit "VALUE" zu vergleichen. Sofern der Wert übereinstimmt, so wird eine 1 zurückgegeben, ansonsten eine 0.

Bsp.: JSON@status.effect@none  
Sucht in diesem Objekt nach dem Feld „status.effect“. Ist der Wert dieses Objektes „none“ so wird das Ergebnis auf 1 ansonsten auf 0 gesetzt. Der gefundene Wert wird umgewandelt, sodass dieser als AI verwendet werden kann.

## Beispiele

- **TEXT@conn-state:**  
Wertet z.B. aus ob die Antwort aus dem Text conn\_state besteht und setzt dementsprechend den Wert auf 1 bzw. auf 0.
- **JSON@twilight.status:**  
Nimmt als Antwort ein JSON Objekt und sucht in diesem Objekt nach dem Feld „twilight.status“ und wandelt diesen Wert um, sodass dieser als AI verwendet werden kann. Gültig sind sämtliche numerischen Werte sowie auch True/False.
- **JSON@temperatures[0].value:**  
Befindet sich im JSON-Objekt ein Array, kann dieser Ausdruck verwendet werden um ein spezifisches Feld auszulesen.
- **JSON@lights.item0.sumstate.value[1]:**  
Ein bestimmter Wert in einem Array kann so ausgelesen werden.
- **JSON@status.effect@none:**  
Nimmt als Antwort ein JSON Objekt und sucht in diesem Objekt nach dem Feld „status.effect“. Ist der Wert dieses Objektes „none“ so wird das Ergebnis auf 1 ansonsten auf 0 gesetzt.
- **TEXT@Hka\_Mw1.sWirkleistung=@VALUE@:**  
Wertet die Zahl nach dem Parameter „Hka\_Mw1.sWirkleistung=“ aus und wandelt diese entsprechend um und kann in einem AI genutzt werden. Bsp.: Hka\_Mw1.sWirkleistung=5,67

## 3.2 GET-Output

Mit dem Typ GET-Output können mit dem HTTP GET Request Werte von my-GEKKO gesendet werden.

Über GET lassen sich auch Formular-Daten übermitteln. Diese werden in codierter Form der URL angehängt. URL und Formular-Daten sind durch ein Fragezeichen (?) voneinander getrennt.

Die Requests werden nur bei Wertänderung gesendet.



### HINWEIS

Die maximale Länge von Name + Request On + Request Off/Parameter darf in der Summe nicht über 250 Zeichen betragen. Pro Request beträgt die maximale Länge 127 Zeichen.

### Anwendungsmöglichkeiten

#### 1. ON/OFF (Ein/Aus)

##### Request On:

Bsp.: `http://192.168.2.27/api/v1/var/lights/item0/scmd?value=1&username=www&password=www`

##### Request Off:

Bsp.: `http://192.168.2.27/api/v1/var/lights/item0/scmd?value=1&username=www&password=www`

#### 2. VALUE (Wert)

Falls kein Request OFF eingetragen wird, so wird immer der Request ON ausgeführt.

### Request On:

**@VALUE@** wird mit dem übergebenen Wert aus AO oder DO als Ganzzahl ersetzt.

Bsp.: Bei dem Wert 60 wird die Lampe Nr. 1 auf 60% Helligkeit gedimmt. Es wird folgender Befehl verwendet, um den Status weiterzugeben:  
`http://192.168.2.27/api/v1/var/lights/item0/scmd?value=D@VALUE@&username=www&password=www`

@VALUE@ ist gleich 60 in unserem Beispiel, daraus folgt dass der gesendete Befehl so aussieht:  
`http://192.168.2.27/api/v1/var/lights/item0/scmd?value=D60&username=www&password=www`

Genauso kann der Platzhalter für Wörter verwendet werden:

**@VALUE:0[true],1[false]@**

Bsp.: Unsere Beispiellampe 1 wird eingeschaltet und sendet damit den Befehl. Es wird folgender Befehl verwendet, um den Status weiterzugeben:  
`http://192.168.2.27/api/v1/var/lights/item0/scmd?value=@VALUE:0[true],1[false]@&username=www&password=www`

@VALUE@ ist gleich 1 in unserem Beispiel, daraus folgt dass Befehl die "1" in "true" umwandelt und folgenden Befehl sendet:  
`http://192.168.2.27/api/v1/var/lights/item0/scmd?value=true&username=www&password=www`



#### HINWEIS

Im Request darf kein ; (Strichpunkt) eingegeben werden, da dieses verwendet wird um Parameter zu trennen.

### 3.3 POST-Output

Mit dem Typ POST-Output können mit dem HTTP POST Request Werte von myGEKKO gesendet werden.

Die POST-Methode funktioniert ähnlich wie die GET-Methode. POST wird jedoch zur Übermittlung von Formular-Daten an ein Programm oder Skript verwendet. Die Daten werden im Entity-Bereich getrennt durch eine Leerzeile vom Header übertragen.

Die Requests werden nur bei Wertänderung gesendet.



#### HINWEIS

Die maximale Länge von Name + Request On + Request Off/Parameter darf in der Summe nicht über 250 Zeichen betragen.

#### Request On (URL):

Bsp.: EBEE Ladestation  
`http://Benutzername:Passwort@IP-Adresse/operator/operator`  
`http://operator:yellow_zone@192.168.2.36/operator/operator`

#### Request OFF/Parameter (Stellt die zu übermittelnden Daten dar):

Die Parameter werden fest eingestellt mit Header *content-type:application/x-www-form-urlencoded* übermittelt.

Bsp.: EBEE Ladestation  
`OperatorCurrentLimit_vehicleif=@VALUE@&SUBMITTYPE=0d`

**@VALUE@** wird mit dem übergebenen Wert aus AO oder DO als Ganzzahl ersetzt

Bsp.: Bei dem Wert 16 = `OperatorCurrentLimit_vehicleif=16&SUBMITTYPE=0d`  
Bei dem Wert 9 = `OperatorCurrentLimit_vehicleif=9&SUBMITTYPE=0d`

**@VALUE:1[true],0[false]**@ wird mit dem übergebenen Wert als Wort ersetzt.



#### HINWEIS

Im Request darf kein ; (Strichpunkt) eingegeben werden, da dieses verwendet wird um Parameter zu trennen.

## 3.4 PUT-Output

Mit dem Typ PUT-Output können mit dem HTTP PUT Request Werte von my-GEKKO gesendet werden.

Diese Methode erlaubt das Erstellen oder Ändern von Dateien auf einem Webserver.

Die Requests werden nur bei Wertänderung gesendet.



### HINWEIS

Die maximale Länge von Name + Request On + Request Off/Parameter darf in der Summe nicht über 250 Zeichen betragen.

### Request On (URL):

Beispiel an Philips Hue:

`http://IP-Adresse Hue Bridge/api/Benutzername/lights/Lampennummer/state`

Die fett geschriebenen Wörter müssen ausgetauscht werden:

`http://192.168.2.49/api/ZqJswGAs76Pf7vwiD8MF5tnvn7bEA5knev 3N7JQ/lights/4/state`

### Request OFF/Parameter:

Stellt die zu übermittelnden Daten dar. Diese werden fest eingestellt mit Header *content-type: application/x-www-form-urlencoded* übermittelt.

Bsp.: Philips Hue - Helligkeit einstellen  
`{"bri": @VALUE@}`

@VALUE@ wird mit dem übergebenen Wert aus AO oder DO als Ganzzahl ersetzt:

Bsp.: Bei dem Wert 50 = `{"bri": 50}` wird die Lampe auf 50% gedimmt  
Bei dem Wert 75 = `{"bri": 75}` wird die Lampe auf 75% gedimmt

Bsp.: Philips Hue - Einschalten  
`{"on": @VALUE:0[false],1[true]@}`

@VALUE:1[true],0[false]@ wird mit dem übergebenen Wert als Wort ersetzt.

Bsp.: Bei dem Wert 1= `{"on": true}` wird die Lampe eingeschaltet  
Bei dem Wert 0= `{"on": false}` wird die Lampe ausgeschaltet



### HINWEIS

Im Request bzw. Parameter darf kein ; (Strichpunkt) eingegeben werden, dieses Zeichen wird in der Eingabe aktuell nicht unterstützt.

## 3.5 GET-Input

Mit dem Typ GET-Input können mit dem HTTP GET Request Werte aus einer HTTP API ausgelesen werden.

Es wird eine Ressource unter Angabe eines URL vom Server/Gerät angefordert. Mit diesem Typ werden nur Daten abgerufen.



### HINWEIS

Die maximale Länge von Name + Request On + Request Off/Parameter darf in der Summe nicht über 250 Zeichen betragen.

Die maximale Länge der Antwort beträgt 1000 Zeichen.

### Request ON = URL:

Bsp:

*myGEKKO API:* `http://192.168.2.30/api/v1/meteo/status?username=www&password=www`

### Für Arrays:

`http://192.168.2.30/api/v1/var/status?username=www&password=www&format=array`

*Dachs Senertec:* `http://glt:@192.168.0.192:8080/getKey?k=Hka_M-w1.sWirkleistung`

### Request OFF/Parameter:

Ist dieses Feld leer, so nimmt Controller an, die Antwort ist bereits eine Zahl und wandelt diese entsprechend um und kann in einem AI genutzt werden.

### Beispiele für mögliche Parameter:

Für weitere Erklärung der Parameter siehe [Erklärung Variablen, Seite 8](#)

#### ■ TEXT@conn-state:

Wertet z.B. aus ob die Antwort aus dem Text conn\_state besteht und setzt dementsprechend den Wert auf 1 bzw. auf 0.

#### ■ JSON@twilight.status:

Nimmt als Antwort ein JSON Objekt und sucht in diesem Objekt nach dem Feld „twilight.status“ und wandelt diesen Wert um, sodass dieser als AI verwendet werden kann. Gültig sind sämtliche numerischen Werte sowie auch True/False.

#### ■ JSON@temperatures[0].value:

Befindet sich im JSON-Objekt ein Array, kann dieser Ausdruck verwendet werden um ein spezifisches Feld auszulesen.

#### ■ JSON@lights.item0.sumstate.value[1]:

Ein bestimmter Wert in einem Array kann so ausgelesen werden.

- **JSON@status.effect@none:**

Nimmt als Antwort ein JSON Objekt und sucht in diesem Objekt nach dem Feld „status.effect“. Ist der Wert dieses Objektes „none“ so wird das Ergebnis auf 1 ansonsten auf 0 gesetzt.

- **TEXT@Hka\_Mw1.sWirkleistung=@VALUE@:**

Wertet die Zahl nach dem Parameter „Hka\_Mw1.sWirkleistung=“ aus und wandelt diese entsprechend um und kann in einem AI genutzt werden. Bsp.: Hka\_Mw1.sWirkleistung=5,67



#### HINWEIS

Im Request bzw. Parameter darf kein ; (Strichpunkt) eingegeben werden, dieses Zeichen wird in der Eingabe aktuell nicht unterstützt.

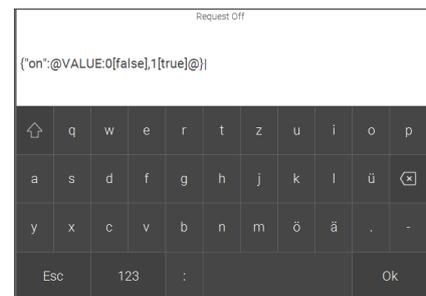
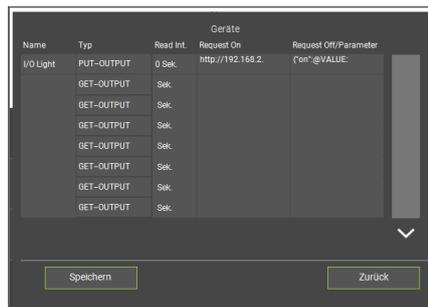
Zum Auslesen der möglichen Werte aus Ihrem myGEKKO Controller, konsultieren Sie das Handbuch zur "Query API". Dort wird die Freischaltung und Zugriff auf die JSON mit den Werten aus Ihrem Controller erklärt.

## 4 IO-Konfiguration Beispiel

### 4.1 Licht

In diesem Beispiel wird beschrieben wie Sie eine Phillips Hue-Lampe ein- und ausschalten können:

1. Melden Sie sich als Konfigurator an, gehen Sie in die Einstellungen ->IO-Stationen und konfigurieren eine neue IO-Station mit dem Typ HTTP/ WebAPI ein. Tippen Sie anschließend auf „Konfiguration>>“.
2. Vergeben Sie dem Befehl einen Namen, dieser muss immer als erstes eingegeben werden.
3. Wählen Sie den HTTP Typ aus. In unserem Fall für die Hue Lampen den Typ PUT-OUTPUT.
4. Geben Sie den Request ON ein. Der Request ist Gerätespezifisch und muss auf Ihr Gerät angepasst werden.
5. Geben Sie den Request OFF/Parameter ein. Der Request ist Gerätespezifisch und muss auf Ihr Gerät angepasst werden.
6. Speichern Sie die Befehle ab.



7. Gehen Sie in das System Licht, legen einen neuen Lichtbaustein an und wechseln in die IO-Konfiguration --> "Ausgänge/Ansteuerungen".
8. Vergeben Sie einen Schalt-Ausgang.
9. Wählen Sie dazu die HTTP IO-Station aus, anschließend tippen Sie auf Ihren konfigurierten Befehl (Device) und senden den PUT Befehl. Wenn Ihr Gerät nicht angezeigt wird, tippen Sie auf den „LOAD“ Button.

Nun können Sie ihre Hue-Lampen ein- und ausschalten.

Diese Konfiguration dient zur Veranschaulichung der HTTP-Befehle und kann auf andere Geräte angepasst werden.

🏠 15:23 Licht ←  
📝 Küche Ausgänge

Schalt-Ausgang	2.1_1	Aus	Typ	Normal	60.00W
Schalt-Rückmeldung	DI	Aus	Wiederholung Aus		
Lichtsystem	Standard	Abgleich..			
Dim-Ausgang 1 (R)	RED 1	0%	0..10V		
Dim-Ausgang 2 (G)	AO	0%	0..10V		
Dim-Ausgang 3 (B)	AO	0%	0..10V		

**Schalt-Ausgang**

STATION	DEVICE	LOAD	REQUEST
2:HTTP	01:1/O Light	▲	SEND->PUT
ONBOARD		▼	
MEMORY	NC	C	Esc
			Ok



*HTTP / Web API*

## **Technisches Handbuch**

myGEKKO | Ekon GmbH

St. Lorenznerstr. 2  
I-39031 Bruneck (BZ)  
Tel. +039 0474 551820  
info@my.gekko.com

**[www.my-gekko.com](http://www.my-gekko.com)**

**Ekon Vertriebs GmbH**  
Fürstenrieder Str. 279a, D-81377 München

**Vertriebsbüro Eislingen**  
Schillerstr. 21, D-73054 Eislingen