

FEMS App REST/JSON schreibend

Version:2023.6.1

Inhaltsverzeichnis

1. Einleitung **2**

2. Installation der App **2**

3. FEMS App REST/JSON schreibend **2**

3.1. Voraussetzungen . . . 2

3.2. Grundlagen REST/JSON . . . 2

3.3. Schreibzugriff . . . 2

3.3.1. Timeout . . . 2

3.3.2. /channel Endpunkt . . . 3

3.3.3. Datenpunkte . . . 3

3.3.4. Beispiel 1 — Wirkleistungsvorgabe: Python . . . 3

3.3.5. Beispiel 2 — Wirkleistungsvorgabe: Talend API Tester . . . 4

4. Kontakt **6**

5. Verzeichnisse **7**

5.1. Abbildungsverzeichnis . . . 7

5.2. Tabellenverzeichnis . . . 8

1. Einleitung

Sehr geehrte Kundin, sehr geehrter Kunde,

vielen Dank, dass Sie sich für die »FEMS App REST/JSON schreibend« entschieden haben. Gerne können Sie uns Ihre Anregungen mitteilen, damit wir die Qualität unserer Produkte noch weiter verbessern können.

2. Installation der App

Mit der Bestellung der »FEMS App REST/JSON schreibend« haben Sie einen 16-stelligen Lizenzschlüssel erhalten. Mittels diesem Lizenzschlüssel können Sie die App eigenständig im EMS App Center einlösen.

Eine Anleitung zur Vorgehensweise finden Sie hier.

3. FEMS App REST/JSON schreibend

Diese Anleitung dient der Beschreibung des Schreibzugriffs auf ein Heckert Solar-Stromspeichersystem mittels REST-/JSON-API. Anschließend wird die Funktionsweise der Schnittstelle erklärt.

3.1. Voraussetzungen

Das auf das Stromspeichersystem zugreifende Gerät (z. B. Notebook/PC) muss direkten Zugriff auf die IP-Adresse des EMS haben — also z. B. im selbem physischen Netzwerk angeschlossen sein.

3.2. Grundlagen REST/JSON

Die REST-/JSON-Schnittstelle ermöglicht den Zugriff auf das EMS im lokalen Netzwerk über eine an REST angelehnte Schnittstelle.

3.3. Schreibzugriff

Diese App stellt eine an REST angelehnte Schnittstelle zur Verfügung, mit der Datenpunkte im System beschrieben werden können.

Diese App ist **nicht** im Standard-Lieferumfang des EMS enthalten. Sie kann jedoch nachträglich jederzeit nachgerüstet werden. Bitte kontaktieren Sie uns, falls Sie eine Nachrüstung wünschen.

Die Verwendung des Schreibzugriffs ist nicht über den Gast-Zugang möglich. Stattdessen ist ein gesonderter Kundenzugang notwendig. Hierfür ist das Passwort "owner" zu verwenden. Der Nutzername kann wie beim Lesezugriff beliebig gewählt werden.

Sämtliche Schreibzugriffe müssen als POST-Requests gesendet werden.

3.3.1. Timeout

Diese App verfügt über einen konfigurierbaren Timeout. Im Standard ist dieser auf 60 Sekunden konfiguriert. Er

sorgt dafür, dass ein Vorgabewert 60 Sekunden lang aktiv bleibt. Sobald ein neuer Vorgabewert geschrieben wird, wird der neue Wert verwendet. Erfolgt kein neuer Vorgabewert innerhalb von 60 Sekunden, fällt die Steuerung auf den nachrangig priorisierten Controller zurück — z. B. Vorgabe einer "0"-Leistung oder Eigenverbrauchsoptimierung.

3.3.2. /channel Endpunkt

Über den Endpunkt `/channel` wird der Zugriff auf einzelne Datenpunkte, sogenannte "Channels", im System ermöglicht.

Die vollständige Adresse des Endpunkts lautet:

http://x:<PASSWORT>@<IP>:80/rest/channel/<KOMPONENTE>/<KANAL>

- `<KOMPONENTE>` ist die ID der Komponente.
- `<KANAL>` ist die ID des Kanals

3.3.3. Datenpunkte

Die folgenden Datenpunkte der Komponente *_ess0* können beschrieben werden:

Datenpunkt	Beschreibung	Einheit
`SetActivePowerEquals`	Vorgabe Be- bzw. Entladeleistung	Watt [W]
`SetReactivePowerEquals`	Vorgabe Blindleistungsvorgabe	VoltAmpere Reaktiv [var]
`SetActivePowerLessOrEquals`	Vorgabe maximaler Entladeleistung	Watt [W]
`SetReactivePowerLessOrEquals`	Vorgabe maximaler Blindleistung	VoltAmpere Reaktiv [var]
`SetActivePowerGreaterOrEquals`	Vorgabe maximaler Beladeleistung	Watt [W]
`SetReactivePowerGreaterOrEquals`	Vorgabe minimaler Blindleistung	VoltAmpere Reaktiv [var]

Table 1. Datenpunkte der Komponente *_ess0*



Die Register für Blindleistungsvorgaben sind aktuell für Home-Systeme nicht nutzbar.



Mehr Informationen zum Channel `SetActivePowerEquals` und anderen Channels zur Leistungsvorgabe finden Sie im Glossar

3.3.4. Beispiel 1 — Wirkleistungsvorgabe: Python

Um z. B. dem ersten Stromspeichersystem (bzw. Stromspeicher-Cluster) eine Entladeleistung von 5 kW vorzugeben, senden Sie einen `POST`-Request an die Adresse `http://192.168.0.23:80/rest/channel/ess0/SetActivePowerEquals` mit der Leistungsvorgabe im *JSON*-Format

```
{
  "value": 5000
}
```



Positive Werte entsprechen einer Speicherentladung - Negative Werte einer Speicherbeladung.

Hierfür kann die *requests*-Bibliothek genutzt werden, die zu Beginn importiert werden muss:

```
import requests

url = 'http://192.168.0.23:80/rest/channel/ess0/SetActivePowerEquals'

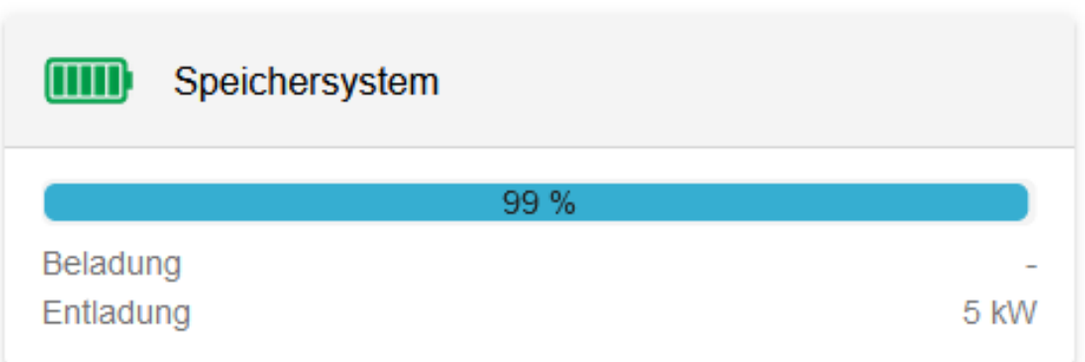
user = 'x'
password = 'owner'

session = requests.Session()
session.auth = (user, password)

data = {"value": 5000}

response = session.post(url, json = data)
response.raise_for_status()
```

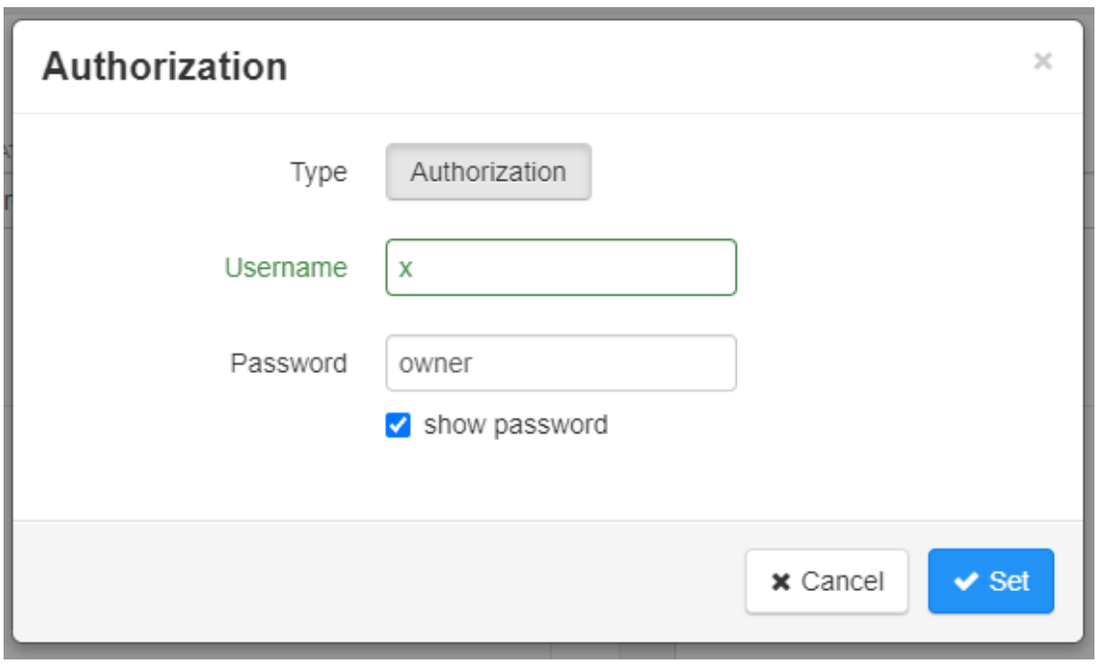
Die korrekte Durchführung des Requests kann über einen anschließenden `GET`-Request oder über das Online-Monitoring (s. unten) überprüft werden.



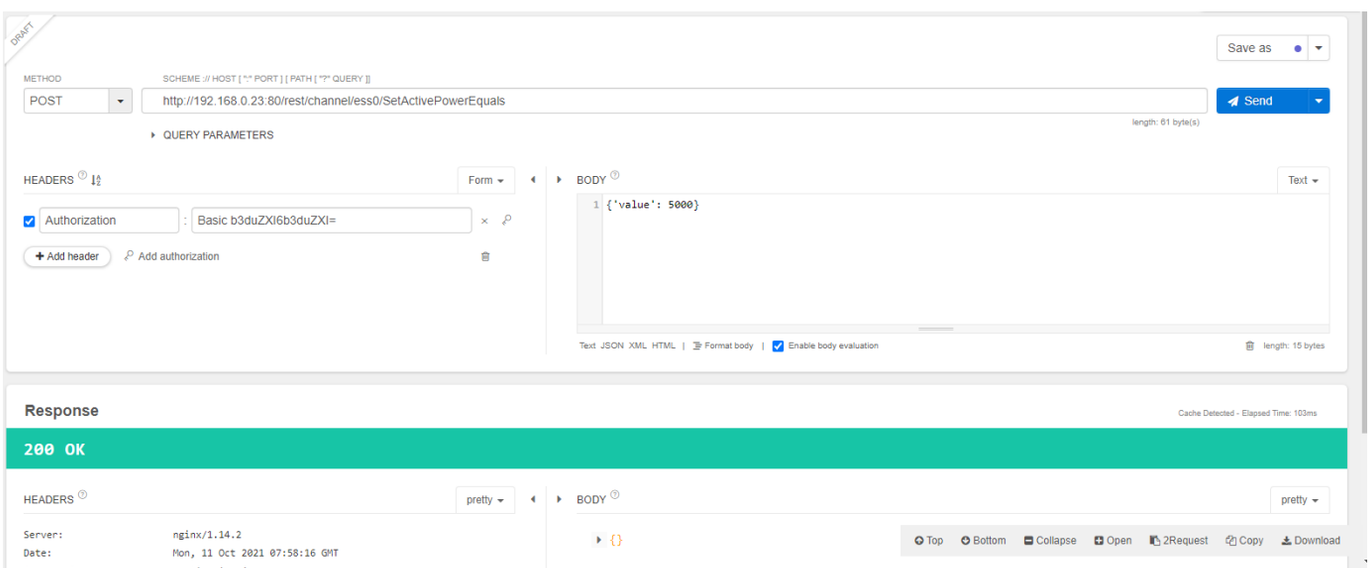
3.3.5. Beispiel 2 — Wirkleistungsvorgabe: Talend API Tester

Talend API Tester ist eine Erweiterung für Google Chrome, die es ermöglicht, REST APIs zu testen.

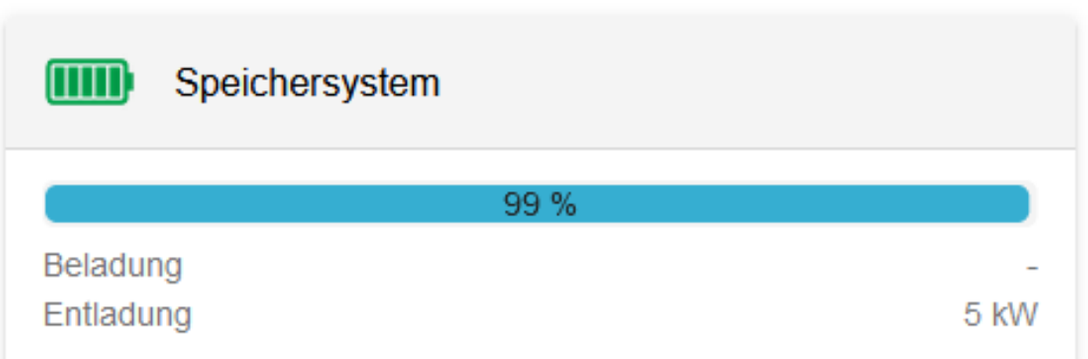
Zunächst muss ein *Authorization*-Header hinzugefügt werden:



Anschließend kann der POST-Request ausgeführt werden.



Die korrekte Durchführung des Requests kann über einen anschließenden GET-Request oder über das Online-Monitoring (s. unten) überprüft werden.



4. Kontakt

Für Unterstützung wenden Sie sich bitte an:

Symphon-E Service

Telefon Service: +49 (0) 371 45 85 68 - 100

E-Mail Service: symphon-e@heckert-solar.com

5. Verzeichnisse

5.1. Abbildungsverzeichnis

5.2. Tabellenverzeichnis

Table 1. Datenpunkte der Komponente _ess0