

fenecon

Benutzerhandbuch – FEMS-App
REST/JSON Lese- und Schreibzugriff

Version 2021.3

Inhalt

1. Einleitung	1
2. Voraussetzungen	1
3. FEMS-App REST/JSON Lese- und Schreibzugriff	2
3.1. Lesezugriff	2
3.2. Schreibzugriff	6
4. Aktivierung der FEMS-App	9
5. Kontakt	10

1. Einleitung

Sehr geehrte Kundin, sehr geehrter Kunde,

vielen Dank, dass Sie sich für die »FEMS-App REST/JSON Lese-/Schreibzugriff« entschieden haben. Gerne können Sie uns Ihre Anregungen mitteilen, damit wir die Qualität unserer Produkte weiterentwickeln können.

2. Voraussetzungen

Für den Einsatz der »FEMS-App REST/JSON Lese-/Schreibzugriff« ist ein FENECON Energiemanagementsystem (FEMS) notwendig.

FEMS wird als Open-Source-Projekt unter dem Namen "OpenEMS" gemeinsam mit vielen weiteren Unternehmen und Instituten in der "OpenEMS Association e.V." entwickelt. Mehr Informationen:

FEMS: <https://fenecon.de/fenecon-fems/>

OpenEMS: <https://www.openems.io>

FEMS ist Produktbestandteil der integrierten FENECON Stromspeichersysteme und weiterer Produktkombinationen aus Batteriewechselrichtern und Batterien. Mehr Informationen dazu finden Sie auf <https://fenecon.de/>

3. FEMS-App REST/JSON Lese- und Schreibzugriff

Die FEMS-Apps REST/JSON Lese- und Schreibzugriff ermöglichen den Zugriff auf das FEMS im lokalen Netzwerk über eine an REST angelehnte Schnittstelle. Es ist dafür notwendig, dass das zugreifende Gerät direkten Zugriff auf die IP-Adresse des FEMS hat – also z. B. im gleichen physischen Netzwerk angeschlossen ist.

3.1. Lesezugriff

Die "FEMS App REST/JSON Lesezugriff" stellt eine an REST angelehnte Schnittstelle zur Verfügung, mit der Datenpunkte im System ausgelesen werden können.



Diese App ist im Standard-Lieferumfang des FEMS enthalten.

Die Basis-Adresse für die REST-Zugriffe lautet <http://<BENUTZER>:<PASSWORT>@<IP>:80/rest>

- **http** ist das Protokoll
- **<BENUTZER>** ist der Benutzername. Da die Authentifizierung lediglich über das Passwort erfolgt, kann hier ein beliebiger Wert (z. B. "x") angegeben werden
- **<PASSWORT>** ist das Passwort des Benutzers. Der Standard "Gast"-Benutzer im FEMS hat das Passwort "**user**"
- **<IP>** ist die IP-Adresse des FEMS
- **80** ist der Port für die REST/JSON-API (optional)

Wenn Ihr FEMS also die lokale IP-Adresse '192.168.0.23' hat, lautet die Basis-Adresse für REST-Zugriffe <http://x:user@192.168.0.23:80/rest>

3.1.1. /channel Endpunkt

Der **/channel** Endpunkt ermöglicht den Zugriff auf einzelne Datenpunkte, sogenannte "Channels", im System.

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**

3.1.2. Datenpunkte

Die folgenden Datenpunkte können ausgelesen werden:

Datenpunkt	Beschreibung	Einheit
_sum/State	Zustand des Systems (0: Ok, 1:Info, 2:Warning, 3:Fault)	
_sum/EssSoc	Ladezustand des Speichers	Prozent [%]
_sum/EssActivePower	Wirkleistung des Speichers	Watt [W]
_sum/GridActivePower	Wirkleistung am Netzanschlusspunkt	Watt [W]
_sum/ProductionActivePower	Wirkleistung Erzeuger	Watt [W]
_sum/ProductionAcActivePower	Wirkleistung AC Erzeuger	Watt [W]
_sum/ProductionDcActualPower	Wirkleistung DC Erzeuger	Watt [W]
_sum/ConsumptionActivePower	Wirkleistung Verbraucher	Watt [W]
_sum/EssActiveChargeEnergy	Energie Speicherbeladung	WattHours [Wh]
_sum/EssActiveDischargeEnergy	Energie Speicherentladung	WattHours [Wh]
_sum/GridBuyActiveEnergy	Energie Netzbezug	WattHours [Wh]
_sum/GridSellActiveEnergy	Energie Netzeinspeisung	WattHours [Wh]
_sum/ProductionActiveEnergy	Energie Erzeugung	WattHours [Wh]
_sum/ProductionAcActiveEnergy	Energie AC Erzeugung	WattHours [Wh]
_sum/ProductionDcActiveEnergy	Energie DC Erzeugung	WattHours [Wh]
_sum/ConsumptionActiveEnergy	Energie Verbraucher	WattHours [Wh]
_sum/EssActivePowerL1	Wirkleistung Phase 1 Speicher	Watt [W]
_sum/EssActivePowerL2	Wirkleistung Phase 2 Speicher	Watt [W]
_sum/EssActivePowerL3	Wirkleistung Phase 3 Speicher	Watt [W]
_sum/GridActivePowerL1	Wirkleistung Phase 1 Netz	Watt [W]
_sum/GridActivePowerL2	Wirkleistung Phase 2 Netz	Watt [W]
_sum/GridActivePowerL3	Wirkleistung Phase 3 Netz	Watt [W]
_sum/ProductionAcActivePowerL1	Wirkleistung Phase 1 Erzeuger	Watt [W]
_sum/ProductionAcActivePowerL2	Wirkleistung Phase 2 Erzeuger	Watt [W]
_sum/ProductionAcActivePowerL3	Wirkleistung Phase 3 Erzeuger	Watt [W]
_sum/ConsumptionActivePowerL1	Wirkleistung Phase 1 Verbraucher	Watt [W]
_sum/ConsumptionActivePowerL2	Wirkleistung Phase 2 Verbraucher	Watt [W]

<code>_sum/ConsumptionActivePowerL3</code>	Wirkleistung Phase 3 Verbraucher	Watt [W]
--	-------------------------------------	----------

3.1.3. Beispiel 1 - Abfrage des Ladezustands: cURL

Das Kommandozeilen-Programm `cURL` ist sowohl unter Windows und Linux vorinstalliert.

Um den Ladezustand des Stromspeichers auszulesen, senden Sie einen `GET`-Request an die Adresse: http://x:user@192.168.0.23:80/rest/channel/_sum/EssSoc

Sie erhalten eine Antwort im JSON-Format:

Windows

Der folgende Befehl speichert die Antwort im JSON-Format in die Datei `out.json`

```
>curl -o out.json http://x:user@192.168.0.23:80/rest/channel/_sum/EssSoc
```

Um den Inhalt der Datei auszugeben, nutzen Sie:

```
>type out.json
```

Ausgabe:

```
{"address": "_sum/EssSoc", "type": "INTEGER", "accessMode": "RO", "text": "", "unit": "%", "value": 99}
```

Den Wert des Ladezustands finden Sie unter `value`. Im Beispiel oben beträgt er 99 %.

Linux

Der folgende Befehl speichert die Antwort im JSON-Format in die Datei `out.json`

```
$curl -o out.json http://x:user@192.168.0.23:80/rest/channel/_sum/EssSoc
```

Um den Inhalt der Datei auszugeben, nutzen Sie:

```
>cat out.json
```

Ausgabe:

```
{"address": "_sum/EssSoc", "type": "INTEGER", "accessMode": "RO", "text": "", "unit": "%", "value": 99}
```

Den Wert des Ladezustands finden Sie unter *value*. Im Beispiel oben beträgt er 99 %.

3.1.4. Beispiel 2 - Abfrage des Ladezustands: Python

Python Versionen für Windows und Linux erhalten Sie hier: <https://www.python.org/downloads/>

Um den Ladezustand des Stromspeichers auszulesen, muss ebenfalls ein **GET**-Request an die Adresse:

http://x:user@192.168.0.23:80/rest/channel/_sum/EssSoc gesendet werden.

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

```
import requests

url = 'http://192.168.0.23:80/rest/channel/_sum/EssSoc'

user = 'x'
password = 'user'

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

response = session.get(url)
response.raise_for_status()
```

Der Befehl liefert eine Antwort im JSON-Format. Diese kann mit dem folgenden Befehl ausgegeben werden:

```
print(response.text)
```

Ausgabe:

```
{"address": "_sum/EssSoc", "type": "INTEGER", "accessMode": "RO", "text": "", "unit": "%", "value": 99}
```

Den Wert des Ladezustands finden Sie unter *value*. Im Beispiel oben beträgt er 99 %.

3.1.5. Beispiel 3 - Abfrage des Ladezustands: 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:

Authorization

Type **Authorization**

Username

Password

show password

✕ Cancel

✓ Set

Anschließend kann der **GET**-Request ausgeführt werden.

The screenshot shows a REST client interface. At the top, the method is set to GET and the URL is `http://192.168.0.23:80/rest/channel/_sum/EssSoc`. The headers section shows an Authorization header with the value `Basic b3duZXI6b3duZXI=`. The response section shows a `200 OK` status and a JSON body: `{ "address": "_sum/EssSoc", "type": "INTEGER", "accessMode": "R0", "text": "", "unit": "%", "value": 99 }`.

Den Wert des Ladezustands finden Sie unter *value*. Im Beispiel oben beträgt er 99 %.

3.2. Schreibzugriff

Die "FEMS App REST/JSON Schreibzugriff" stellt eine an **REST W** angelegte Schnittstelle zur Verfügung, mit der Datenpunkte im System beschrieben werden können.



Diese App ist **nicht** im Standard-Lieferumfang des FEMS enthalten. Sie kann jedoch nachträglich jederzeit nachgerüstet werden. Kontaktieren Sie uns hierfür!



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 Nutzernamen kann wie beim Lesezugriff beliebig gewählt werden.

Sämtliche Schreibzugriffe müssen als **POST**-Requests gesendet werden.

3.2.1. Timeout

Die "FEMS App REST/JSON-API Schreibzugriff" 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.

3.2.2. /channel Endpunkt

Genau wie bei der "FEMS App REST/JSON Lesezugriff", wird über den Endpunkt `/channel` 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](#)

3.2.3. Datenpunkte

Die folgenden Datenpunkte können beschrieben werden:

Datenpunkt	Beschreibung	Einheit
<code>ess0/SetActivePowerEquals</code>	Wirkleistungsvorgabe	Watt [W]
<code>ess0/SetReactivePowerEquals</code>	Blindleistungsvorgabe	VoltAmpereReactive [VAR]
<code>ess0/SetActivePowerLessOrEquals</code>	(Maximale) Wirkleistungsvorgabe	Watt [W]
<code>ess0/SetReactivePowerLessOrEquals</code>	(Maximale) Blindleistungsvorgabe	VoltAmpereReactive [VAR]
<code>ess0/SetActivePowerGreaterOrEquals</code>	(Minimale) Wirkleistungsvorgabe	Watt [W]
<code>ess0/SetReactivePowerGreaterOrEquals</code>	(Minimale) Wirkleistungsvorgabe	VoltAmpereReactive [VAR]



Mehr Informationen zum Channel `SetActivePowerEquals` und anderen Channels zur Leistungsvorgabe finden Sie im [Glossar](#)

3.2.4. 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
}
```



Eine Entladung des Speichers wird durch einen positiven, eine Beladung durch einen negativen Wert umgesetzt.

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'

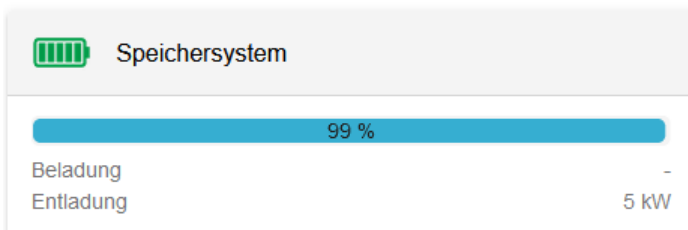
user = 'x'
password = 'owner'

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 das Online-Monitoring (s. unten) überprüft werden.



3.2.5. 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:

Authorization

Type: Authorization

Username: x

Password: owner

show password

Cancel Set

Anschließend kann der **POST**-Request ausgeführt werden.

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://192.168.0.23:80/rest/channel/ess0/SetActivePowerEquals
- Headers:** Authorization: Basic b3duZXI6b3duZXI=
- Body:** 1 { 'value': 5000 }
- Response:** 200 OK
- Response Headers:** Server: nginx/1.14.2, Date: Mon, 11 Oct 2021 07:58:16 GMT

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

The screenshot shows a battery status widget with the following information:

- Speichersystem** (Battery System)
- Charge Level:** 99 %
- Beladung (Loading):** -
- Entladung (Discharging):** 5 kW

4. Aktivierung der FEMS-App

Falls Sie die FEMS-App direkt mit Ihrem Speicher bestellt haben, wurde sie bereits auf dem FEMS vorkonfiguriert und ist sofort aktiv. Falls Sie die FEMS-App nachrüsten, muss das FEMS noch per Fernwartung konfiguriert werden. Kontaktieren Sie uns dazu bitte unter service@fenecon.de und geben Sie bitte Ihre FEMS-Nr. (z. B. „fems123“) an, sowie um welche App es sich handelt.

5. Kontakt

Für Unterstützung wenden Sie sich bitte an:

FENECON GmbH

Brunnwiesenstraße 4

94469 Deggendorf

Telefon Service: 0991-648800-33

E-Mail Service: service@fenecon.de