

# API-Dokumentation EVSE-WiFi 2.0 / smartWB

Stand: 19.07.2021

## Inhalt

1. getParameters .....	2
2. getLog .....	4
3. setStatus .....	5
4. setCurrent .....	5
5. clearRfid .....	5
6. interruptCp .....	6
7. doReboot .....	6
8. evseHost .....	7

## 1. getParameters

Der Aufruf **getParameters** gibt essentielle Daten zum Status der Wallbox und dem aktuellen Ladevorgang zurück.

Parameter	Beschreibung	Datentyp
vehicleState	Fahrzeugstatus (1: bereit   2: Fahrzeug angeschlossen   3: Fahrzeug lädt   5: Fehler)	integer
evseState	EVSE Status (true: Ladestation freigeschaltet   false: Ladestation gesperrt)	boolean
maxCurrent	Maximal möglicher Ladestrom (abhängig vom PP-Widerstand und dem konfigurierten maximalen Ladestrom – je nachdem welcher Wert niedriger ist)	integer
actualCurrent	Konfigurierter Ladestrom in A (z.B. 15 -> 15A)	integer
actualCurrentMA*	Konfigurierter Ladestrom in hundertstel A (z.B. <u>1550</u> -> 15,50A)	integer
actualPower	Aktuelle Ladeleistung in kW	float
duration	Dauer des Ladevorgangs in ms	integer
alwaysActive	Always Active Mode (true/false)	boolean
lastActionUser	Name des Benutzers, der die letzte Aktivität durchgeführt hat (Aktivieren/Deaktivieren)	string
lastActionUID	UID des Benutzers, der die letzte Aktivität durchgeführt hat (Aktivieren/Deaktivieren)	string
energy	Geladene Energie des aktuellen Ladevorgangs in kWh	float
mileage	Geladene Energie des aktuellen Ladevorgangs in km (abhängig vom konfigurierten Stromverbrauch des Fahrzeugs)	float
meterReading	Zählerstand	float
currentP1	Strom in A (L1)	float
currentP2	Strom in A (L2)	float
currentP3	Strom in A (L3)	float
voltageP1	Spannung in V (L1)	float
voltageP2	Spannung in V (L2)	float
voltageP3	Spannung in V (L3)	float
useMeter	Stromzähler konfiguriert (true/false)	boolean
RFIDUID	Zuletzt erkannte RFID-UID (kann mit /clearRFID manuell zurückgesetzt werden; wird überschrieben, sobald eine neue RFID-Auslesung erfolgt)	string

\* bei smartWB seit Ende Mai 2021 mit EVSE Firmware ab v18 (GUI: Status -> EVSE-Register 1005) – bei früheren Hardware-Versionen wird dieser Parameter nicht zurückgegeben.

Beispiel:

```
GET http://192.168.4.1/getParameters
```

Rückgabe:

```
{
  "type": "parameters",
  "list": [
    {
      "vehicleState": 3,
      "evseState": false,
      "maxCurrent": 16,
      "actualCurrent": 10,
      "actualCurrentMA": 1000,
      "actualPower": 2.251,
      "duration": 1821000,
      "alwaysActive": false,
      "lastActionUser": "GUI",
      "lastActionUID": "GUI",
      "energy": 9.52,
      "mileage": 82.3,
      "meterReading": 54.35,
      "currentP1": 9.78,
      "currentP2": 0,
      "currentP3": 0,
      "voltageP1": 228.28,
      "voltageP2": 231.85,
      "voltageP3": 232.07,
      "useMeter": true,
      "RFIDUID": "0a1b2c3d"
    }
  ]
}
```

## 2. getLog

Ausgabe des Logfiles im JSON-Format

Parameter	Beschreibung	Datentyp
uid	UID des RFID-Benutzers, der den Ladevorgang initiiert hat. Bei einer Aktivierung, welche nicht per RFID ausgeführt wurde gilt: GUI: <u>GUI</u>   Taster: <u>Button</u>   API: <u>remote</u>   Fahrzeug: <u>vehicle</u>	string
username	Vergebener RFID-Benutzername, der den Ladevorgang initiiert hat. Bei einer Aktivierung, welche nicht per RFID ausgeführt wurde gilt: GUI: <u>GUI</u>   Taster: <u>Button</u>   API: <u>remote</u>   Fahrzeug: <u>vehicle</u>	string
timestamp	Unix Timestamp zu Beginn des Ladevorgangs	integer
duration	Dauer des Ladevorgangs in Millisekunden	integer
energy	Geladene Energie in kWh	float
price	Zum Beginn des Ladevorgangs hinterlegter Energiepreis	float
reading*	Zählerstand zu Beginn des Ladevorgangs	float

\*ab Firmware v2.2.0 (Für Logeinträge, welche mit früheren Firmwareversionen erstellt wurden, existiert dieser Parameter nicht)

Beispiel:

```
GET http://192.168.4.1/getLog
```

Rückgabe:

```
{
  "type": "latestlog",
  "list": [
    {
      "uid": "GUI",
      "username": "GUI",
      "timestamp": 1609264401,
      "duration": 201699,
      "energy": 0.22,
      "price": 30,
      "reading": 252.496
    },
    {
      "uid": "GUI",
      "username": "GUI",
      "timestamp": 1609264620,
      "duration": 222506,
      "energy": 0.24,
      "price": 30,
      "reading": 252.256
    }
  ]
}
```

### 3. setStatus

Über den Befehl **setStatus** kann ein Ladevorgang initiiert oder beendet werden, sofern EVSE-WiFi im Normal Mode agiert. Im Always Active oder Remote Mode kann der Ladevorgang durch eine Stromvorgabe von 0A (siehe **setCurrent**) beendet werden.

Parameter	Beschreibung	Datentyp
active	true: Ladevorgang starten false: Ladevorgang beenden	Boolean

Beispiel:

```
GET http://192.168.4.1/setStatus?active=true
```

Startet den Ladevorgang.

### 4. setCurrent

Über den Befehl **setCurrent** kann der konfigurierte Ladestrom manuell gesetzt werden. Bei Verwendung des Always Active oder Remote Modes kann durch eine Stromvorgabe von 0 der Ladevorgang beendet werden.

Parameter	Beschreibung	Datentyp
current	Vorgegebener Ladestrom in A oder hundertstel A*.	integer

\* Vorgaben in hundertstel A werden erst ab EVSE-Firmware 18 korrekt umgesetzt. Vorgaben in hundertstel A bei Verwendung einer EVSE-Firmware < 18 werden auf volle A abgerundet

Beispiele:

```
GET http://192.168.4.1/setCurrent?current=10
```

Setzt den zulässigen Ladestrom auf 10A

```
GET http://192.168.4.1/setCurrent?current=1050
```

Setzt den zulässigen Ladestrom auf 10,50A, falls EVSE-Firmware >= 18 verwendet wird. Bei EVSE-Firmware <18 wird der zulässige Ladestrom auf 10,0A gesetzt.

### 5. clearRfid

Der Aufruf **clearRfid** initialisiert den Parameter „RFIDUID“, welcher über den Aufruf **getParameters** ausgelesen werden kann (siehe auch **getParameters**).

Beispiel:

```
GET http://192.168.4.1/clearRfid
```

Initialisiert den Wert des Parameters „RFIDUID“, welcher über den Aufruf **getParameters** zurückgegeben wird.

## 6. interruptCp

Unterbricht das CP-Signal (relaisgesteuert) für 3 Sekunden.

Beispiel:

```
GET http://192.168.4.1/interruptCp
```

Unterbricht das CP-Signal für 3 Sekunden.

## 7. doReboot

Führt einen Systemneustart durch.

Parameter	Beschreibung	Datentyp
reboot	true: Neustart wird unmittelbar durchgeführt	Boolean

Beispiel:

```
GET http://192.168.4.1/doReboot?reboot=true
```

Führt einen Systemneustart durch.

## 8. evseHost

Gibt Netzwerk- und weitere Konfigurationswerte zurück.

Parameter	Beschreibung	Datentyp
ssid	Konfigurierte SSID	string
dns	Konfigurierter DNS-Server	string
mac	MAC-Adresse der Wallbox	string
ip	IP-Adresse der Wallbox	string
netmask	Konfigurierte Subnetzmaske	string
gateway	Konfiguriertes Gateway	string
uptime	Betriebszeit seit Einschalten der Wallbox in Sekunden	integer
opMode*	Betriebsmodus: normal   alwaysActive   remote	string
firmware*	Firmware-Version	string

\*ab Firmware v2.2.5

**Hinweis:** Ab Firmware v2.2.5 wird dieser API-Aufruf auch beantwortet, wenn in den Einstellungen die API grundsätzlich deaktiviert ist.

Beispiel:

```
GET http://192.168.4.1/evseHost
```

```
{
  "type": "evseHost",
  "list": [
    {
      "ssid": "EVSE-WiFi",
      "dns": "192.168.4.1",
      "mac": "94:B9:7E:69:F0:D1",
      "ip": "192.168.4.1",
      "netmask": "255.255.255.0",
      "gateway": "0.0.0.0",
      "uptime": 1960,
      "opMode": "normal",
      "firmware": "2.2.5"
    }
  ]
}
```