Inhaltsverzeichnis

openWB simpleAPI HTTP ... 3
Features ... 3
Verwendung ... 3
Lesende Anfragen ... 3
Schreibende Anfragen ... 4
Alle verfügbaren Parameter ... 5
Chargepoint (Ladepunkt) Parameter ... 5
Counter (Zähler) Parameter ... 6
Battery (Batterie) Parameter ... 7
PV (Photovoltaik) Parameter ... 7
Consumer (Verbraucher) Parameter ... 7
IO (Ein-/Ausgänge) Parameter ... 8
Schreibbare Parameter ... 9
Komponenten-Übersicht (list_components) ... 10
Auto-ID Feature ... 10
Total-Abfragen ... 10
Authentifizierung ... 11
Mit Bearer Token ... 11
Mit Username/Passwort ... 11
Ausgabeformate ... 11
JSON-Format (Standard) ... 11
Raw-Format ... 12
Beispiele ... 12
Vollständiges Monitoring-Dashboard ... 12
Ladevorgang steuern ... 12
Instant Charging komplett konfigurieren ... 12
Fahrzeug-Management ... 13
Template-basierte Überwachung ... 13
Neue Features in dieser Version ... 13
Template-basierte Parameter ... 13
Erweiterte Chargepoint-Daten ... 14
Instant Charging Control ... 14
Vehicle Management ... 14
Konfigurationszugriff ... 14
simpleAPI für MQTT ... 14
Features ... 14
Installation ... 15
Topic-Transformation (Lesezugriff) ... 15
Basis-Transformation ... 15
JSON-Aufspaltung ... 15
Tuple/Array-Aufspaltung (1-basiert) ... 15
Lademodus auslesen ... 15
ID-Vereinfachung ... 16
Total-Abfragen ... 16
SoC-Werte (Ladezustand) ... 16
Schreiboperationen (Set-Topics) ... 18
1. Lademodus setzen ... 18
2. Ladestrom für Sofortladen ... 18

3. Minimaler EV SoC im PV Laden 18
4. Minimaler Dauerstrom für PV Laden 19
5. Maximaler Preis für ECO Laden 19
6. Ladepunkt sperren 19
7. Speicherbeachtung konfigurieren 19
8. Instant Charging Limit-Typ 20
9. Instant Charging SoC-Grenze 20
10. Instant Charging Energiegrenze 20
11. PV Charging Limit-Typ 21
12. PV Charging SoC-Grenze 21
13. PV Charging Energiegrenze 21
Intelligente Features 21
Überwachte Komponenten 22

Es gibt 2 simpleAPI:

- simpleAPI (HTTP)
- simpleAPI (MQTT)

---

openWB simpleAPI HTTP

Features

- **65+ Parameter** - Umfassende Unterstützung für Chargepoint, Counter, Battery und PV
- **Auto-ID Feature** - Automatische Auswahl der niedrigsten verfügbaren ID
- **Template-basierte Settings** - Erweiterte openWB-Template-Manipulation
- **Raw & JSON Output** - Flexible Ausgabeformate
- **GET & POST Support** - Beide HTTP-Methoden unterstützt
- **Instant Charging Control** - Vollständige Sofortladen-Konfiguration
- **Vehicle Management** - Fahrzeugzuordnung und SoC-Management
- **Consumer (Verbraucher) Unterstützung** – Lesezugriff auf alle Verbraucher-Werte
- **Komponenten-Übersicht** – `list_components` liefert alle konfigurierten IDs je Typ auf einen Blick
- **PV-Ladelimit** – `pv_charging_limit`/`pv_charging_amount`/`pv_charging_soc` als Stopp-Limit für den Lademodus „PV-Laden“, analog zum bereits vorhandenen Stopp-Limit für „Sofortladen“

Verwendung

Lesende Anfragen

Alle Daten eines Ladepunkts:

```
curl "http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_all=3"
```

Auto-ID Feature - niedrigste verfügbare ID:

```
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_all=auto"
# oder ohne Angabe einer ID:
curl "http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_all="
```

Spannung Phase 1 (Raw-Ausgabe):

```
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_voltage_p1=
3&raw=true"
```

Batterie-Daten:

```
curl "http://IPADRESSE/openWB/simpleAPI/simpleapi.php?battery=2"
```

Kombination mehrerer Werte:

```
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_all=3&batte
ry=2"
```

Neue Parameter abrufen:

```
# Tägliche Energie und Frequenz
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_daily_impor
ted=3&get_chargepoint_frequency=3"

# RFID und Konfigurationsdaten
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_rfid=3&get_
chargepoint_config_name=3"

# Template-Daten (Instant Charging & PV)
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_instant_cha
rging_current=3&get_chargepoint_pv_charging_min_current=3"

# Leistungsfaktoren
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_power_facto
rs=3&raw=false"
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_power_facto
r_p1=3&raw=true"
```

Schreibende Anfragen

Lademodus ändern:

```
curl -X POST -d "set_chargemode=pv&chargepoint_nr=3" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"Auto
```

Auto-ID Feature beim Schreiben:

```
# Schreibt an den Ladepunkt mit der niedrigsten ID
curl -X POST -d "set_chargemode=pv" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
```

Ladestrom setzen:

```
curl -X POST -d "chargecurrent=8.23&chargepoint_nr=3" \
```

```
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
```

Batterie-Modus setzen:

```
curl -X POST -d "bat_mode=min_soc_bat_mode" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
```

reservierte BatterieLeistung setzen:

```
curl -X POST -d "bat_power_reserve=2500" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
```

Wert in Watt

Instant Charging konfigurieren:

```
# Energielimit setzen
curl -X POST -d "instant_charging_limit=amount&chargepoint_nr=3" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"

# Energiemenge setzen (25.5 kWh)
curl -X POST -d "instant_charging_amount=25.5&chargepoint_nr=3" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"

# SoC-Limit setzen (80%)
curl -X POST -d "instant_charging_soc=80&chargepoint_nr=3" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
```

Fahrzeug-Management:

```
# Fahrzeug 1 zu Ladepunkt 3 zuordnen
curl -X POST -d "vehicle=1&chargepoint_nr=3" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"

# Manuellen SoC auf 75% setzen
curl -X POST -d "manual_soc=75&chargepoint_nr=3" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
```

Alle verfügbaren Parameter

Chargepoint (Ladepunkt) Parameter

Parameter	Beschreibung	Beispiel
get_chargepoint_all	Alle Daten	?get_chargepoint_all=2
get_chargepoint_voltages	Alle Spannungen	?get_chargepoint_voltages=2
get_chargepoint_voltage_p1/p2/p3	Spannung einzelne Phase	?get_chargepoint_voltage_p1=2
get_chargepoint_currents	Alle Ströme	?get_chargepoint_currents=2
get_chargepoint_current_p1/p2/p3	Strom einzelne Phase	?get_chargepoint_current_p1=2

Parameter	Beschreibung	Beispiel
get_chargepoint_powers	Alle Leistungen	?get_chargepoint_powers=2
get_chargepoint_power	Gesamtleistung	?get_chargepoint_power=2
get_chargepoint_imported	Gesamtenergie geladen	?get_chargepoint_imported=2
get_chargepoint_exported	Gesamtenergie zurückgespeist	?get_chargepoint_exported=2
get_chargepoint_daily_imported	Täglicher Import	?get_chargepoint_daily_imported=2
get_chargepoint_daily_exported	Täglicher Export	?get_chargepoint_daily_exported=2
get_chargepoint_frequency	Netzfrequenz	?get_chargepoint_frequency=2
get_chargepoint_rfid	RFID-Tag	?get_chargepoint_rfid=2
get_chargepoint_rfid_timestamp	RFID-Zeitstempel	?get_chargepoint_rfid_timestamp=2
get_chargepoint_evse_current	EVSE-Strom	?get_chargepoint_evse_current=2
get_chargepoint_power_factors	Alle Leistungsfaktoren	?get_chargepoint_power_factors=2
get_chargepoint_power_factor_p1/p2/p3	Leistungsfaktor einzelne Phase	?get_chargepoint_power_factor_p1=2
get_chargepoint_config_name	Ladepunkt-Name	?get_chargepoint_config_name=2
get_chargepoint_connected_vehicle_name	Fahrzeug-Name	?get_chargepoint_connected_vehicle_name=2
get_chargepoint_charge_template_name	Lade-Profil Name	?get_chargepoint_charge_template_name=2
get_chargepoint_instant_charging_current	Instant Charging Strom	?get_chargepoint_instant_charging_current=2
get_chargepoint_pv_charging_min_current	PV Min-Strom	?get_chargepoint_pv_charging_min_current=2
get_chargepoint_soc	Batteriestand (%)	?get_chargepoint_soc=2
get_chargepoint_state_str	Status-Text	?get_chargepoint_state_str=2
get_chargepoint_fault_str	Fehler-Text	?get_chargepoint_fault_str=2
get_chargepoint_fault_state	Fehler-Code	?get_chargepoint_fault_state=2
get_chargepoint_phases_in_use	Verwendete Phasen	?get_chargepoint_phases_in_use=2
get_chargepoint_plug_state	Stecker angeschlossen	?get_chargepoint_plug_state=2
get_chargepoint_charge_state	Ladevorgang aktiv	?get_chargepoint_charge_state=2
get_chargepoint_chargemode	Lademodus	?get_chargepoint_chargemode=2

Counter (Zähler) Parameter

Parameter	Beschreibung	Beispiel
get_counter	Alle Zähler-Daten	?get_counter=0
get_counter_voltages	Alle Spannungen	?get_counter_voltages=0
get_counter_voltage_p1/p2/p3	Spannung einzelne Phase	?get_counter_voltage_p1=0
get_counter_currents	Alle Ströme	?get_counter_currents=0
get_counter_current_p1/p2/p3	Strom einzelne Phase	?get_counter_current_p1=0
get_counter_powers	Alle Leistungen	?get_counter_powers=0
get_counter_power	Gesamtleistung	?get_counter_power=0
get_counter_power_factors	Leistungsfaktoren	?get_counter_power_factors=0
get_counter_imported	Gesamtenergie bezogen	?get_counter_imported=0
get_counter_exported	Gesamtenergie eingespeist	?get_counter_exported=0
get_counter_daily_imported	Tagesenergie bezogen	?get_counter_daily_imported=0
get_counter_daily_exported	Tagesenergie eingespeist	?get_counter_daily_exported=0
get_counter_frequency	Netzfrequenz	?get_counter_frequency=0
get_counter_fault_str	Fehler-Text	?get_counter_fault_str=0
get_counter_fault_state	Fehler-Code	?get_counter_fault_state=0

Battery (Batterie) Parameter

Parameter	Beschreibung	Beispiel
battery	Alle Batterie-Daten	?battery=2
get_battery_power	Batterie-Leistung	?get_battery_power=2
get_battery_soc	Batterie-Ladestand	?get_battery_soc=2
get_battery_currents	Batterie-Ströme	?get_battery_currents=2
get_battery_imported	Batterie geladen	?get_battery_imported=2
get_battery_exported	Batterie entladen	?get_battery_exported=2
get_battery_daily_imported	Tagesenergie geladen	?get_battery_daily_imported=2
get_battery_daily_exported	Tagesenergie entladen	?get_battery_daily_exported=2
get_battery_fault_str	Fehler-Text	?get_battery_fault_str=2
get_battery_fault_state	Fehler-Code	?get_battery_fault_state=2

PV (Photovoltaik) Parameter

Parameter	Beschreibung	Beispiel
pv	Alle PV-Daten	?pv=5
get_pv_power	PV-Leistung	?get_pv_power=5
get_pv_currents	PV-Ströme	?get_pv_currents=5
get_pv_exported	PV-Gesamtertrag	?get_pv_exported=5
get_pv_daily_exported	PV-Tagesertrag	?get_pv_daily_exported=5
get_pv_monthly_exported	PV-Monatsertrag	?get_pv_monthly_exported=5
get_pv_yearly_exported	PV-Jahresertrag	?get_pv_yearly_exported=5
get_pv_fault_str	Fehler-Text	?get_pv_fault_str=5
get_pv_fault_state	Fehler-Code	?get_pv_fault_state=5

Consumer (Verbraucher) Parameter

Parameter	Beschreibung	
get_consumer_all	Alle Verbraucher-Daten	?get_consumer_all=5
get_consumer_power	Aktuelle Leistung	?get_consumer_power=5
get_consumer_currents	Ströme (alle Phasen)	?get_consumer_currents=5
get_consumer_voltages	Spannungen (alle Phasen)	?get_consumer_voltages=5
get_consumer_powers	Leistungen (alle Phasen)	get_consumer_powers=5
get_consumer_imported	Gesamtenergie bezogen	?get_consumer_imported=5
get_consumer_exported	Gesamtenergie zurückgespeist	?get_consumer_exported=5
get_consumer_daily_imported	Tagesenergie bezogen	?get_consumer_daily_imported=5
get_consumer_phases_in_use	Verwendete Phasen	?get_consumer_phases_in_use=5
get_consumer_charge_state	Verbraucher aktiv	?get_consumer_charge_state=5
get_consumer_state_str	Status-Text	?get_consumer_state_str=5
get_consumer_fault_str	Fehler-Text	?get_consumer_fault_str=5
get_consumer_fault_state	Fehler-Code	?get_consumer_fault_state=5
get_consumer_usage_type	Verwendungsart (z. B. meter_only)	?get_consumer_usage_type=5

Alternative Schreibweisen für „alle Daten“, analog zu `battery`/`pv`: `consumer` und `get_consumer` funktionieren wie `get_consumer_all`.

Beispiel:

```
curl "http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_consumer_all=5"
```

```
{
  "consumer_5": {
    "usage_type": "meter_only",
    "power": 34.5,
    "currents": [0.79, 0.71, 0.7],
    "voltages": [231, 233.7, 232.5],
    "powers": [24.8, 3.4, 6.3],
    "imported": 2140.94,
    "exported": 0,
    "daily_imported": 11.33,
    "phases_in_use": 0,
    "charge_state": false,
    "state_str": "Messwerte des Verbrauchers werden erfasst.",
    "fault_str": "Kein Fehler.",
    "fault_state": 0
  }
```

IO (Ein-/Ausgänge) Parameter

Parameter	Beschreibung	Beispiel
`get_io_output_all`	Alle Ausgänge (digital + analog) eines IO-Geräts	`?get_io_output_all=2`
`get_io_output`	Einzelner Ausgangsstatus (Name über `io_output`)	`?get_io_output=2&io_output=DO1`

Hinweise:

- Für Auto-ID kann `auto` oder ein leerer Wert verwendet werden (z. B. `?get_io_output_all=auto`).
- Bei `get_io_output` kann optional `io_output_type` angegeben werden: `digital` oder `analog`.
- Ohne `io_output` liefert `get_io_output` den gleichen Inhalt wie `get_io_output_all`.

Beispiele (Lesen):

```
# Alle Ausgänge (digital + analog), IO-ID explizit
curl "http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_io_output_all=2"

# Alle Ausgänge per Auto-ID
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_io_output_all=auto"

# Einzelnen digitalen Ausgang lesen
curl
```

```
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_io_output=2&io_output=DO1"

# Einzelnen Addon-Ausgang lesen (z. B. LED1)
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_io_output=2&io_output=LED1"

# Einzelnen analogen Ausgang lesen (Typ explizit)
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_io_output=2&io_output=AO1&io_output_type=analog"
```

Beispiele (Schreiben):

```
# Digitalen Ausgang setzen (0/1 oder true/false)
curl -X POST -d "set_io_output=1&io_nr=2&io_output=DO1" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"

# Addon-Ausgang setzen (z. B. LED1) per Auto-ID
curl -X POST -d "set_io_output=true&io_nr=auto&io_output=LED1" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"

# Analogen Ausgang setzen (Typ optional explizit)
curl -X POST -d
"set_io_output=45.5&io_nr=2&io_output=AO1&io_output_type=analog" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
```

Schreibbare Parameter

Parameter	Werte	Beschreibung
set_chargemode	instant, pv, eco, stop, target	Lademodus setzen
chargecurrent	0, 6-32	Ladestrom in Ampere
minimal_pv_soc	0-100	Min. SoC für PV-Laden
minimal_permanent_current	0, 6-32	Min. Dauerstrom für PV
max_price_eco	-100 bis 200	Max. Preis für ECO-Laden
chargepoint_lock	0, 1	Ladepunkt sperren
bat_mode	min_soc_bat_mode, ev_mode, bat_mode	Batterie-Modus
bat_power_reserve	Wert in Watt	reservierte Batterieleistung
instant_charging_limit	none, amount, soc	Instant Charging Limittyp
instant_charging_amount	≥ 0 (kWh)	Instant Charging Energielimit
instant_charging_soc	0-100 (%)	Instant Charging SoC-Limit
pv_charging_limit	none, amount, soc	PV-Laden Limittyp
pv_charging_amount	≥ 0 (kWh)	PV-Laden Energielimit
pv_charging_soc	0-100 (%)	PV-Laden SoC-Limit
vehicle	≥ 0 (Vehicle ID)	Fahrzeug zuordnen
manual_soc	0-100 (%)	Manueller SoC setzen

Komponenten-Übersicht (list_components)

Parameter, mit dem sich alle konfigurierten Komponenten-IDs je Typ auf einen Blick abfragen lassen - nützlich, um vor einer Integration (z. B. Home Assistant, ioBroker, eigene Skripte) herauszufinden, welche IDs überhaupt existieren, ohne jeden Typ einzeln zu raten.

```
curl "http://IPADRESSE/openWB/simpleAPI/simpleapi.php?list_components=all"
```

```
{
  "chargepoint": [0, 2, 3],
  "battery": [2],
  "pv": [1, 3],
  "counter": [0, 4],
  "consumer": [5],
  "io": []
}
```

Typen ohne konfigurierte Geräte liefern eine leere Liste statt eines Fehlers.

Auto-ID Feature

Das Auto-ID Feature ist eine Erweiterung der openWB SimpleAPI. Wenn keine ID angegeben wird oder `auto` verwendet wird, wird automatisch die niedrigste verfügbare ID verwendet:

```
# Diese Aufrufe sind equivalent:
curl "http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_all="
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_all=auto"

# Findet automatisch den Ladepunkt mit der niedrigsten ID
curl -X POST -d "chargemode=pv"
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
```

Total-Abfragen

Bei ausgewählten Parametern kann statt einer ID der Wert `total` verwendet werden. Dann wird der Wert über alle vorhandenen IDs der Komponente summiert.

Beispiele:

```
curl "http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_pv_power=total"
curl "http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_pv_imported=total"
curl "http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_pv_exported=total"
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_pv_monthly_exported=tot
al"
```

```
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_pv_yearly_exported=tota
l"

curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_power=total
"
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_imported=to
tal"
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_exported=to
tal"
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_monthly_exp
orted=total"
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_yearly_expo
rted=total"
```

Authentifizierung

Mit Bearer Token

```
curl -H "Authorization: Bearer your-token" \
     -X POST -d "chargemode=pv&chargepoint_nr=3" \
     "https://IPADRESSE/openWB/simpleAPI/simpleapi.php"
```

Mit Username/Passwort

```
curl -X POST \
     -d "username=admin" \
     -d "password=admin123" \
     -d "chargemode=pv" \
     -d "chargepoint_nr=3" \
     "https://IPADRESSE/openWB/simpleAPI/simpleapi.php"
```

Ausgabeformate

JSON-Format (Standard)

```
{
  "chargepoint_3": {
    "power": 0,
    "voltages": [237.79, 0, 0],
```

```
    "currents": [0, 0, 0],
    "state_str": "Nicht bereit"
  }
}
```

Raw-Format

Für Kompatibilität mit der originalen SimpleAPI kann das Raw-Format verwendet werden:

```
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_power=3&raw
=true"
# Ausgabe: 0
```

Beispiele

Vollständiges Monitoring-Dashboard

```
# Alle wichtigen Werte in einem Aufruf
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_all=auto&ba
ttery=auto&pv=auto&get_counter=0"
```

Ladevorgang steuern

```
# PV-Laden aktivieren
curl -X POST -d "set_chargemode=pv&chargepoint_nr=auto" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"

# Ladestrom auf 16A setzen
curl -X POST -d "chargecurrent=16&chargepoint_nr=auto" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"

# Batterie-Modus auf EV-Mode setzen
curl -X POST -d "bat_mode=ev_mode" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
```

Instant Charging komplett konfigurieren

```
# Sofortladen mit 20 kWh Limit einrichten
curl -X POST -d "instant_charging_limit=amount&chargepoint_nr=3" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
curl -X POST -d "instant_charging_amount=20&chargepoint_nr=3" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
```

```
curl -X POST -d "set_chargemode=instant&chargepoint_nr=3" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"

# Sofortladen mit SoC-Limit (80%) einrichten
curl -X POST -d "instant_charging_limit=soc&chargepoint_nr=3" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
curl -X POST -d "instant_charging_soc=80&chargepoint_nr=3" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
curl -X POST -d "set_chargemode=instant&chargepoint_nr=3" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
```

Fahrzeug-Management

```
# Fahrzeug zuweisen und SoC setzen
curl -X POST -d "vehicle=1&chargepoint_nr=3" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
curl -X POST -d "manual_soc=65&chargepoint_nr=3" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"

# Automatisches Fahrzeug für Auto-ID Ladepunkt
curl -X POST -d "vehicle=2" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
curl -X POST -d "manual_soc=50" \
     "http://IPADRESSE/openWB/simpleAPI/simpleapi.php"
```

Template-basierte Überwachung

```
# Alle Template-Daten in einem Aufruf
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_charge_template_name=3&get_chargepoint_instant_charging_current=3&get_chargepoint_pv_charging_min_current=3"

# Konfiguration und Status kombiniert
curl
"http://IPADRESSE/openWB/simpleAPI/simpleapi.php?get_chargepoint_config_name=3&get_chargepoint_connected_vehicle_name=3&get_chargepoint_rfid=3"
```

Neue Features in dieser Version

Template-basierte Parameter

- **`get_chargepoint_instant_charging_current`** - Liest Sofortlade-Stromstärke aus Template
- **`get_chargepoint_pv_charging_min_current`** - Liest PV-Mindestrom aus Template
- **`get_chargepoint_charge_template_name`** - Template-Name

Erweiterte Chargepoint-Daten

- **`get_chargepoint_daily_imported/exported`** - Tägliche Energiebilanz
- **`get_chargepoint_frequency`** - Netzfrequenz-Überwachung
- **`get_chargepoint_rfid/rfid_timestamp`** - RFID-Event-Tracking
- **`get_chargepoint_power_factors`** - Leistungsfaktor-Monitoring
- **`get_chargepoint_evse_current`** - EVSE-Hardware-Status

Instant Charging Control

- **`instant_charging_limit`** - Limittyp konfigurieren (none/amount/soc)
- **`instant_charging_amount`** - Energielimit in kWh setzen
- **`instant_charging_soc`** - SoC-Limit in Prozent setzen

Vehicle Management

- **`vehicle`** - Fahrzeug-ID einem Ladepunkt zuordnen
- **`manual_soc`** - Manuellen Batteriestand setzen

Konfigurationszugriff

- **`get_chargepoint_config_name`** - Ladepunkt-Name aus Konfiguration
- **`get_chargepoint_connected_vehicle_name`** - Fahrzeug-Name aus Status

---

simpleAPI für MQTT

Dieser Dienst ist dafür gedacht, komplexe MQTT-Topic-Strukturen in eine vereinfachte API umzuwandeln. Der Dienst greift `openWB/*` Topics auf und republiziert diese unter `openWB/simpleAPI/*` mit aufgeteilten JSON/Tuple-Werten und ID-Vereinfachung.

Features

- **Wert-Änderungserkennung**: Publiziert nur wenn sich Werte tatsächlich ändern
- **JSON/Tuple-Aufspaltung**: Zerlegt komplexe Datenstrukturen in einzelne Topics
- **ID-Vereinfachung**: Niedrigste IDs werden auch ohne ID-Suffix verfügbar gemacht
- **Retained Messages**: Alle Nachrichten werden mit retain=true publiziert
- **TLS/SSL Support**: Optional verschlüsselte Verbindungen
- **Schreiboperationen**: Vereinfachte Steuerung von Ladeparametern

Installation

Ist Teil der standard openWB Installation ab 2.1.9

Topic-Transformation (Lesezugriff)

Basis-Transformation

Originale Topics werden von `openWB/` zu `openWB/simpleAPI/` transformiert.

Wichtig: Der `/get/` Teil wird aus den Topics entfernt:

```
Original: openWB/chargepoint/0/get/power
Wird zu: openWB/simpleAPI/chargepoint/0/power
```

JSON-Aufspaltung

```
Original: openWB/chargepoint/0/config/template {"id": 0, "name": "Test",
"max_current": 32}
Wird zu:
- openWB/simpleAPI/chargepoint/0/config/template/id → 0
- openWB/simpleAPI/chargepoint/0/config/template/name → Test
- openWB/simpleAPI/chargepoint/0/config/template/max_current → 32
```

Tuple/Array-Aufspaltung (1-basiert)

```
Original: openWB/counter/3/get/voltages [232.4, 0.0, 0.0]
Wird zu:
- openWB/simpleAPI/counter/3/voltages/1 → 232.4
- openWB/simpleAPI/counter/3/voltages/2 → 0.0
- openWB/simpleAPI/counter/3/voltages/3 → 0.0
```

Lademodus auslesen

```
openWB/simpleAPI/chargepoint/3/chargemode
oder (niedrigste ID):
openWB/simpleAPI/chargepoint/chargemode
```

Der Lademodus wird aus `openWB/chargepoint/3/get/connected_vehicle/config/chargemode` transformiert und vereinfacht zu `chargemode` am Ladepunkt.

ID-Vereinfachung

Die ID eines Ladepunktes als auch von Komponenten ist im Webinterface unter status einsehbar.

Die niedrigste ID jeder Komponente wird auch ohne ID verfügbar gemacht:

```
Original: openWB/counter/0/get/power 1042.4
Wenn 0 die niedrigste counter-ID ist:
- openWB/simpleAPI/counter/0/power → 1042.4
- openWB/simpleAPI/counter/power → 1042.4
```

Total-Abfragen

Bei ausgewählten Parametern kann statt einer ID der Wert `total` verwendet werden. Dann wird der Wert über alle vorhandenen IDs der Komponente summiert.

Beispiele:

```
"openWB/simpleAPI/pv/total/#"
"openWB/simpleAPI/chargepoint/total/#"
```

SoC-Werte (Ladezustand)

Es gibt verschiedene SoC-Werte die verfügbar gemacht werden:

```
# Aktueller EV-SoC (aus connected_vehicle JSON)
Original: openWB/chargepoint/0/get/connected_vehicle/soc
JSON: {"soc": 82.8, "fault_state": 0, "range": null, ...}
Wird zu: openWB/simpleAPI/chargepoint/0/soc/soc → 82.8
Die übrigen Werte sind unter soc/* zu finden.

# Direkter Pro-SoC Wert
Original: openWB/chargepoint/0/get/soc
Wird zu: openWB/simpleAPI/chargepoint/0/pro_soc → 75

# Pro-SoC Zeitstempel
Original: openWB/chargepoint/0/get/soc_timestamp
Wird zu: openWB/simpleAPI/chargepoint/0/pro_soc_timestamp → 1762777573
```

Erklärung:

- **soc**: Der aktuelle Ladezustand des verbundenen Elektrofahrzeugs (extrahiert aus connected_vehicle JSON)

- **pro_soc**: Direkter SoC-Wert vom Ladepunkt

- **pro_soc_timestamp**: Zeitstempel der letzten SoC-Aktualisierung

Beispielausgabe eines Ladepunktes:

```
openWB/simpleAPI/chargepoint/13/manual_lock False
openWB/simpleAPI/chargepoint/13/evse_current 0
openWB/simpleAPI/chargepoint/13/vehicle_name Kona (Daniel)
openWB/simpleAPI/chargepoint/13/chargemode scheduled_charging
openWB/simpleAPI/chargepoint/13/state_str "Keine Ladung, da kein Auto
angesteckt ist."
openWB/simpleAPI/chargepoint/13/charge_state False
openWB/simpleAPI/chargepoint/13/phases_in_use 1
openWB/simpleAPI/chargepoint/13/plug_state False
openWB/simpleAPI/chargepoint/13/power 0
openWB/simpleAPI/chargepoint/13/currents/1 0.0
openWB/simpleAPI/chargepoint/13/currents/2 0.0
openWB/simpleAPI/chargepoint/13/currents/3 0.0
openWB/simpleAPI/chargepoint/13/voltages/1 233.83
openWB/simpleAPI/chargepoint/13/voltages/2 235.37
openWB/simpleAPI/chargepoint/13/voltages/3 235.27
openWB/simpleAPI/chargepoint/13/fault_str "Kein Fehler."
openWB/simpleAPI/chargepoint/13/fault_state 0
openWB/simpleAPI/chargepoint/13/daily_imported 0.0
openWB/simpleAPI/chargepoint/13/daily_exported 0.0
openWB/simpleAPI/chargepoint/13/rfid null
openWB/simpleAPI/chargepoint/13/rfid_timestamp null
openWB/simpleAPI/chargepoint/13/imported 5367626.95
openWB/simpleAPI/chargepoint/13/exported 0
openWB/simpleAPI/chargepoint/13/powers/1 0.0
openWB/simpleAPI/chargepoint/13/powers/2 0.0
openWB/simpleAPI/chargepoint/13/powers/3 0.0
openWB/simpleAPI/chargepoint/13/frequency 49.93
openWB/simpleAPI/chargepoint/13/power_factors/1 1.0
openWB/simpleAPI/chargepoint/13/power_factors/2 1.0
openWB/simpleAPI/chargepoint/13/power_factors/3 1.0
openWB/simpleAPI/chargepoint/13/serial_number "21207688"
openWB/simpleAPI/chargepoint/13/soc null
openWB/simpleAPI/chargepoint/13/soc_timestamp null
openWB/simpleAPI/chargepoint/13/vehicle_id null
openWB/simpleAPI/chargepoint/13/error_timestamp null
openWB/simpleAPI/chargepoint/13/charging_current 0
openWB/simpleAPI/chargepoint/13/charging_power 0
openWB/simpleAPI/chargepoint/13/charging_voltage 0
openWB/simpleAPI/chargepoint/13/version "2.1.9-Alpha.3"
openWB/simpleAPI/chargepoint/13/evse_signaling null
openWB/simpleAPI/chargepoint/13/max_discharge_power null
openWB/simpleAPI/chargepoint/13/max_charge_power null
openWB/simpleAPI/revision 2
```

ab revsion 2

```
# Minimaler Permanentstrom für PV-Laden
openWB/simpleAPI/chargepoint/0/minimal_permanent_current → 6
```

```
# Minimaler EV-SoC für PV-Laden
openWB/simpleAPI/chargepoint/0/minimal_pv_soc → 20

# Maximaler Preis für ECO-Laden
openWB/simpleAPI/chargepoint/0/max_price_eco → 0.25

# Instant-Charging Limit-Typ
openWB/simpleAPI/chargepoint/0/instant_charging_limit → soc

# Instant-Charging SoC-Grenze (in %)
openWB/simpleAPI/chargepoint/0/instant_charging_limit_soc → 80

# Instant-Charging Energiegrenze (in kWh)
openWB/simpleAPI/chargepoint/0/instant_charging_limit_amount → 25
```

Schreiboperationen (Set-Topics)

1. Lademodus setzen

Beschreibung: Lademodus am Ladepunkt ändern

Topic: `openWB/simpleAPI/set/chargepoint/3/chargemode` oder `openWB/simpleAPI/set/chargepoint/chargemode` (niedrigste ID)

Gültige Werte: `instant`, `pv`, `eco`, `stop`, `target`

Beispiel:

```
mosquitto_pub -t "openWB/simpleAPI/set/chargepoint/3/chargemode" -m "pv"
```

2. Ladestrom für Sofortladen

Beschreibung: Ladestrom für das Sofort laden

Topic: `openWB/simpleAPI/set/chargepoint/3/chargecurrent`

Wert: Stromstärke in Ampere (z.B. „16“)

Beispiel:

```
mosquitto_pub -t "openWB/simpleAPI/set/chargepoint/3/chargecurrent" -m "16"
```

3. Minimaler EV SoC im PV Laden

Beschreibung: Minimaler EV SoC im PV Laden

Topic: `openWB/simpleAPI/set/chargepoint/3/minimal_pv_soc`

Wert: SoC in Prozent (z.B. „20“)

Beispiel:

```
mosquitto_pub -t "openWB/simpleAPI/set/chargepoint/3/minimal_pv_soc" -m "20"
```

4. Minimaler Dauerstrom für PV Laden

Beschreibung: Minimaler Dauerstrom für das PV Laden

Topic: `openWB/simpleAPI/set/chargepoint/3/minimal_permanent_current`

Wert: Stromstärke in Ampere (z.B. „6“)

Beispiel:

```
mosquitto_pub -t
"openWB/simpleAPI/set/chargepoint/3/minimal_permanent_current" -m "6"
```

5. Maximaler Preis für ECO Laden

Beschreibung: Maximaler Preis für das ECO Laden

Topic: `openWB/simpleAPI/set/chargepoint/3/max_price_eco`

Wert: Preis als Dezimalzahl (z.B. „0.25“)

Beispiel:

```
mosquitto_pub -t "openWB/simpleAPI/set/chargepoint/3/max_price_eco" -m
"0.25"
```

6. Ladepunkt sperren

Beschreibung: Sperrt den Ladepunkt

Topic: `openWB/simpleAPI/set/chargepoint/3/chargepoint_lock`

Werte: `true/false`

Beispiel:

```
mosquitto_pub -t "openWB/simpleAPI/set/chargepoint/3/chargepoint_lock" -m
"true"
```

7. Speicherbeachtung konfigurieren

Beschreibung: Konfiguriert die Speicherbeachtung

Topic: `openWB/simpleAPI/set/bat_mode`

Gültige Werte: `min_soc_bat_mode`, `ev_mode`, `bat_mode`

Beispiel:

```
mosquitto_pub -t "openWB/simpleAPI/set/bat_mode" -m "ev_mode"
```

reservierte Batterieleistung setzen

Topic: `openWB/simpleAPI/set/bat_power_reserve`

Gültige Werte: int Watt

8. Instant Charging Limit-Typ

Beschreibung: Setzt den Typ der Ladebegrenzung für das Sofortladen

Topic: `openWB/simpleAPI/set/instant_charging_limit`

Gültige Werte: `none`, `soc`, `amount`

Beispiel:

```
mosquitto_pub -t "openWB/simpleAPI/set/instant_charging_limit" -m "soc"
```

9. Instant Charging SoC-Grenze

Beschreibung: Setzt die SoC-Grenze für das Sofortladen (aktiv wenn Limit-Typ „soc“ ist)

Topic: `openWB/simpleAPI/set/instant_charging_limit_soc`

Wert: Integer 0-100 (Prozent)

Beispiel:

```
mosquitto_pub -t "openWB/simpleAPI/set/instant_charging_limit_soc" -m "80"
```

10. Instant Charging Energiegrenze

Beschreibung: Setzt die Energiegrenze für das Sofortladen (aktiv wenn Limit-Typ „amount“ ist)

Topic: `openWB/simpleAPI/set/instant_charging_limit_amount`

Wert: Integer 1-50 (kWh)

Hinweis: Wird intern automatisch in Wh umgerechnet (Eingabewert × 1000)

Beispiel:

```
mosquitto_pub -t "openWB/simpleAPI/set/instant_charging_limit_amount" -m
"25"
```

11. PV Charging Limit-Typ

Beschreibung: Setzt den Typ der Ladebegrenzung für das PV-Laden

Topic: `openWB/simpleAPI/set/pv_charging_limit`

Gültige Werte: `none`, `soc`, `amount`

Beispiel:

```
mosquitto_pub -t "openWB/simpleAPI/set/pv_charging_limit" -m "soc"
```

12. PV Charging SoC-Grenze

Beschreibung: Setzt die SoC-Grenze für das PV-Laden (aktiv wenn Limit-Typ „soc“ ist)

Topic: `openWB/simpleAPI/set/pv_charging_limit_soc`

Wert: Integer 0-100 (Prozent)

Beispiel:

```
mosquitto_pub -t "openWB/simpleAPI/set/pv_charging_limit_soc" -m "80"
```

13. PV Charging Energiegrenze

Beschreibung: Setzt die Energiegrenze für das PV-Laden (aktiv wenn Limit-Typ „amount“ ist)

Topic: `openWB/simpleAPI/set/pv_charging_limit_amount`

Wert: Integer 1-50 (kWh)

Hinweis: Wird intern automatisch in Wh umgerechnet (Eingabewert × 1000)

Beispiel:

```
mosquitto_pub -t "openWB/simpleAPI/set/pv_charging_limit_amount" -m "25"
```

Intelligente Features

- **Automatisches ID-Management**: Wenn keine ID angegeben wird, wird die niedrigste verfügbare ID verwendet

- **JSON-Template-Caching**: Charge-Templates werden automatisch zwischengespeichert
- **Werte-Mapping**: Einfache Werte werden zu internen openWB-Werten gemappt
- **Fehlerbehandlung**: Ungültige Werte werden abgefangen und geloggt
- **Logging**: Alle Schreiboperationen werden ausführlich geloggt

Überwachte Komponenten

Der Daemon überwacht folgende openWB-Komponenten:

- **bat**: Batteriespeicher (Leistung, SoC, Import/Export)
- **pv**: Photovoltaik-Anlagen (Leistung, Ertrag)
- **chargepoint**: Ladepunkte (Status, Konfiguration, Messwerte)
- **counter**: Energiezähler (Leistung, Verbrauch, Netzwerte)
- **consumer**: Verbraucher (Leistung, Status, Fehlerzustand)

Verbraucher-Topics werden wie bat/pv/counter generisch transformiert (keine Sonderbehandlung wie bei chargepoint nötig, d. h. `/get/` bleibt - anders als bei Chargepoint - im Topic-Pfad erhalten), z. B.:

```
Original: openWB/consumer/5/get/power
Wird zu:  openWB/simpleAPI/consumer/5/get/power
```

Bei niedrigster Consumer-ID zusätzlich ohne ID-Suffix verfügbar:

```
openWB/simpleAPI/consumer/get/power
```

Beispiel:

```
mosquitto_sub -h IPADRESSE -p 1884 -t "openWB/simpleAPI/consumer/#" -v
```

```
openWB/simpleAPI/consumer/5/get/power 35.2
openWB/simpleAPI/consumer/5/get/state_str Messwerte des Verbrauchers werden
erfasst.
openWB/simpleAPI/consumer/5/get/fault_str Kein Fehler.
openWB/simpleAPI/consumer/5/get/currents/1 0.8
openWB/simpleAPI/consumer/5/get/currents/2 0.73
openWB/simpleAPI/consumer/5/get/currents/3 0.73
openWB/simpleAPI/consumer/5/config/max_power 5000
openWB/simpleAPI/consumer/5/usage/type meter_only
```

(Nested JSON-Objekte wie `config`, `usage`, `module` werden vollständig in Einzel-Topics aufgespalten; Arrays 1-basiert, siehe bestehender Abschnitt „JSON-Aufspaltung“/„Tuple/Array- Aufspaltung“.)

From:
https://wiki.openwb.de/ - **openWB GmbH & Co. KG**

Permanent link:
https://wiki.openwb.de/doku.php?id=openwb:vc:2.3.x:simpleapi

Last update: **2026/10/28 21:53**