

---

# home-assistant-ocpp

unknown

May 17, 2024



# CONTENTS

<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	Install HACS . . . . .	3
1.2	Install the OCPP Repository . . . . .	3
1.3	Add the OCPP Integration . . . . .	3
1.4	Configure the Central System . . . . .	4
1.4.1	Host address and port . . . . .	4
1.4.2	Secure Connection . . . . .	4
1.4.3	Measurands . . . . .	4
1.5	Add the entities to your Dashboard . . . . .	4
1.6	Configure your Charger . . . . .	5
1.7	Start Charging . . . . .	6
<b>2</b>	<b>Supported devices</b>	<b>9</b>
2.1	ABB Terra AC-W7-G5-R-0 . . . . .	9
2.2	ABB Terra AC-W11-G5-R-0 . . . . .	9
2.3	ABB Terra AC-W22-T-0 . . . . .	9
2.4	Alfen - Eve Single Pro-line . . . . .	9
2.5	Alfen - Eve Single S-line . . . . .	9
2.6	CTEK Chargestorm Connected 2 . . . . .	9
2.7	Etrek - Inch Pro . . . . .	9
2.8	EVBox Elvi . . . . .	10
2.9	EVLink Wallbox Plus . . . . .	10
2.10	Evnex E Series & X Series Charging Stations . . . . .	10
2.11	Garo Entity Pro . . . . .	10
2.12	MaXpeedingrods Ev Charger . . . . .	10
2.13	Simpson & Partners . . . . .	10
2.14	United Chargers Inc. - Grizzl-E . . . . .	10
2.15	V2C Trydan . . . . .	10
2.16	Vestel EVC04-AC22SW . . . . .	10
2.17	Wallbox Pulsar & Copper SB . . . . .	10
2.18	Others . . . . .	10
<b>3</b>	<b>User guide</b>	<b>13</b>
3.1	Installing the OCPP Integration . . . . .	13
3.2	Installing HACS (Home Assistant Community Store) . . . . .	13
3.3	Configuring the Central System . . . . .	14
3.4	Useful Entities for Wallbox Pulsar Plus . . . . .	15
3.4.1	Metrics . . . . .	15
3.4.2	Diagnostics . . . . .	16
3.4.3	Controls . . . . .	16

3.5	Useful Entities for EVBox Elvi . . . . .	16
3.5.1	Metrics . . . . .	16
3.5.2	Diagnostics . . . . .	16
3.5.3	Controls . . . . .	16
3.6	Useful Entities and Workarounds for United Chargers Grizzl-E . . . . .	16
3.6.1	Metrics . . . . .	17
3.6.2	Diagnostics . . . . .	17
3.6.3	Controls . . . . .	17
3.7	Useful Entities for Vestel EVC-04 Wallboxes . . . . .	17
3.7.1	Metrics . . . . .	17
3.7.2	Diagnostics . . . . .	18
3.7.3	Controls . . . . .	18
3.7.4	OCPP Compatibility Issues . . . . .	18
<b>4</b>	<b>Support</b>	<b>19</b>
<b>5</b>	<b>Debugging</b>	<b>21</b>
<b>6</b>	<b>Development</b>	<b>25</b>



This is a Home Assistant integration for Electric Vehicle chargers that support the Open Charge Point Protocol.



## INSTALLATION

### 1.1 Install HACS

- If you have not yet installed HACS, go get it at <https://hacs.xyz> and walk through the installation and configuration.

### 1.2 Install the OCPP Repository

- In Home Assistant, select HACS / Integrations / + Explore & add repositories.

A blue rounded rectangular button with a white plus icon on the left and the text 'EXPLORE & ADD REPOSITORIES' in white capital letters.

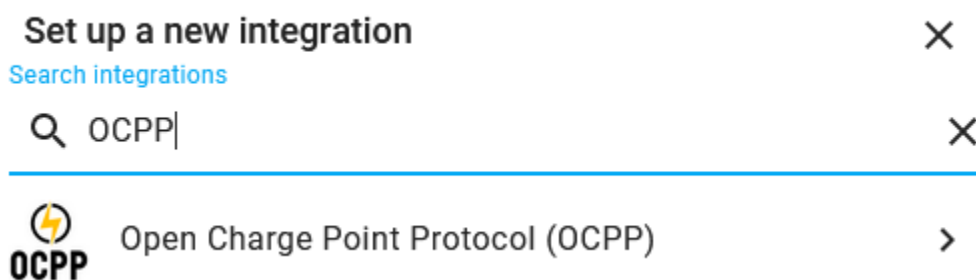
- Search for 'OCPP' and install the repository.

### 1.3 Add the OCPP Integration

- In Home Assistant, select Configuration / Integrations / Add Integration.

A blue rounded rectangular button with a white plus icon on the left and the text 'ADD INTEGRATION' in white capital letters.

- Search for 'OCPP' and add the integration.



## 1.4 Configure the Central System

### 1.4.1 Host address and port

- The default host address '0.0.0.0' will listen to all interfaces on your home assistant server.
- The default port number is 9000 but can be changed for your needs.

### 1.4.2 Secure Connection

If you are using [Let's Encrypt](#), [Duck DNS](#) or other add-on that enables secure HTTPS for your Home Assistant instance, you can get a secure WSS connection for OCPP. To use a secure connection:

- Enable the option *Secure connection*
- Provide the pathways to your HA's SSL certificate and key files. These are typically located in the /config or /ssl folder, and typically named fullchain.pem and privkey.pem respectively.
- If you provide incorrect pathways, the integration will fail to setup with no clear indication of why.

If you do not use HTTPS for your Home Assistant instance:

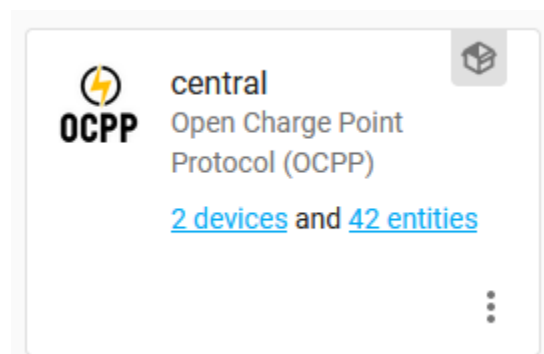
- Disable the option *Secure connection*
- *Path to SSL certificate/key* will be ignored.

### 1.4.3 Measurands

- Most chargers only support a subset of all possible measurands. This depends most on the Feature profiles that are supported by the charger.
- The integration will autodetect the supported measurands when the charger connects.

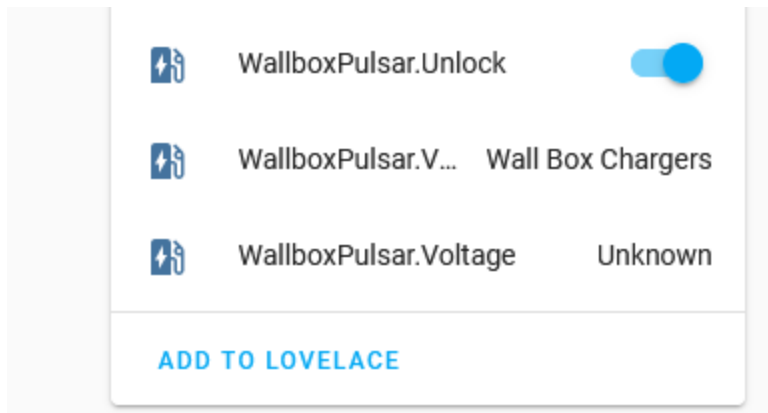
## 1.5 Add the entities to your Dashboard

- On the OCPP integration, click on devices to navigate to your Charge Point device.



- At the bottom of the Entities panel, click on 'Add to Lovelace' to add the entities to your dashboard.





- An entity will have the value 'Unavailable' until the charger successfully connects.
- An entity will have the value 'Unknown' until its value has been read from the charger.

## 1.6 Configure your Charger

- Configure your charger to use the OCPP websocket of your Central System (e.g. `ws://homeassistant.local:9000`). This is charger specific, so consult your manual.
- Some chargers require the protocol section '`ws://`' to be removed, or require the url to end with a '`/`'.
- If you have configured *Secure connection* in previous step, you should use '`wss://`'
- Some chargers require the url to be specified as an IP address, i.e. '`192.168.178.1:9000`'
- You may need to reboot your charger before the changes become effective.

Device detail

OCPP

☒

WEBSOCKET CONNECTION

url

ws://homeassistant.local:9000

charge point identity

WallboxPulsar

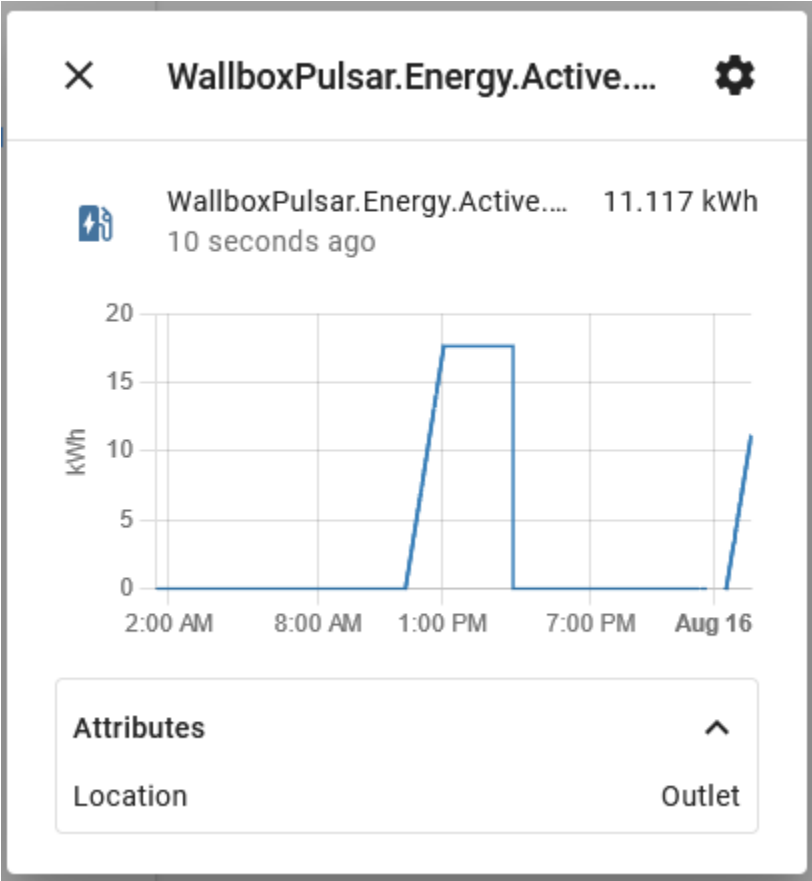
Password

☒ I have read and accepted the [Terms and Conditions](#)

Save

## 1.7 Start Charging

- Use the charge control switch to start the charging process.





## SUPPORTED DEVICES

All OCPP 1.6j compatible devices should be supported, but not every device offers the same level of functionality. So far, we've tried:

### 2.1 ABB Terra AC-W7-G5-R-0

### 2.2 ABB Terra AC-W11-G5-R-0

### 2.3 ABB Terra AC-W22-T-0

### 2.4 Alfen - Eve Single Pro-line

### 2.5 Alfen - Eve Single S-line

### 2.6 CTEK Chargestorm Connected 2

Jonas Karlsson has written a [getting started guide](#) for connecting CTEK Chargestorm Connected 2.

### 2.7 Etrek - Inch Pro

To allow a custom OCPP server such as HA to set up a transaction ID, it is necessary to set under Users > Charging Authorization the authorization type to either `Central system only` or `Charger whitelist and central system` otherwise the OCPP integration won't match transactions and it won't report some meter values such as session time.

## 2.8 EVBox Elvi

## 2.9 EVLink Wallbox Plus

## 2.10 Evnex E Series & X Series Charging Stations

(Ability to configure a custom OCPP server such as HA is being discontinued)

## 2.11 Garo Entity Pro

## 2.12 MaXpeedingrods Ev Charger

## 2.13 Simpson & Partners

All basic functions work properly

## 2.14 United Chargers Inc. - Grizzl-E

(has some defects in OCPP implementation, which can be worked around. See [User Guide](#) section in Documentation for details.)

## 2.15 V2C Trydan

## 2.16 Vestel EVC04-AC22SW

## 2.17 Wallbox Pulsar & Copper SB

The Wallbox Pulsar and Copper SB have been verified. In the OCPP-config, leave the password field empty.

## 2.18 Others

When a charger is not listed as a supported charger it simply means that it has not been reported to work. Whether it will work or not in practice really depends on whether it is compliant with the OCPP standard. Some vendors claim their device is compliant without bothering to do a compliance test, because that takes time and costs money!

When it is fully compliant, then it should work out of the box, since the ocpp integration is designed to work for fully compliant chargers. Any issues should be reported, and we will do our best to analyze them. In some cases modifications or workarounds may be needed. As long as these workarounds do not break compliance to the OCPP standard they can be added to this repository. Otherwise, we urge you to request your vendor to update their firmware to make their device OCPP compliant.

You can always make your own fork of this repository to solve issues for a specific device that are not OCPP compliant. However, we will not integrate these type of changes into this repository, because that may prevent other chargers to work.





## 3.1 Installing the OCPP Integration

Follow the steps listed in [README.md](#) to get started. Below are some additional notes which may save you some time.

## 3.2 Installing HACS (Home Assistant Community Store)

Installation of the HACS integration is a pre-requisite before you can install OCPP. However, it's worth noting that HACS brings a lot of baggage along with it, which is annoying, but this is the price to pay for using a 3rd party repository installer such as HACS. Having said that, once it's up and running, HACS stays out of the way unless you need to Redownload or Remove OCPP.

The 'baggage' referred to above, is every single repository available through HACS. As you can imagine, this adds up to a huge amount of data being downloaded from the Github servers, and they get upset about it, displaying `Rate Limit` error messages. You will see these error messages whenever you install HACS, but don't worry, the rate limit will reset after a few hours and HACS will be installed. It's worth remembering never to remove HACS unless there is no other way to achieve whatever it is you're wanting to do. Each time you reinstall, you'll be in for a wait of several hours so it's best avoided unless there is no other alternative.

### 3.3 Configuring the Central System

#### OCPP Configuration



If you need help with the configuration have a look here: <https://github.com/lbbrhzn/ocpp>

Central system host address

0.0.0.0

Central system port number

9000



Central system identity

central

Charge point identity

charger

Meter interval (seconds)

60



SUBMIT

The `Central system identity` shown above with a default of `central` can be anything you like. Whatever is entered in that field will be used as a device identifier in Home Assistant (HA), so it's probably best to avoid spaces and punctuation symbols, but otherwise, enter anything you like.

The `Charge point identity` shown above with a default of `charger` is a little different. Whatever you enter in that field will determine the prefix of all Charger entities added to Home Assistant (HA). My recommendation is that it's best left at the default of `charger`. If you put anything else in that field, it will be used as the prefix for all Charger entities added to HA during installation, however, new entities subsequently added in later version releases sometimes revert to the default prefix, regardless of what was entered during installation. So you end up with a mixture of different prefixes which can be avoided simply by leaving `Charge point identity` set to the default of `charger`.

## OCPP Measurands



Select which measurand(s) should be shown in Home Assistant.

- ☒ Active energy imported from the grid
- ☐ Reactive energy imported from the grid
- ☐ Active energy imported from the grid during last interval
- ☐ Reactive energy imported from the grid during last interval
- ☐ Instantaneous active power imported by EV

Measurands (according to OCPP terminology) are actually metrics provided by the charger. Each charger supports a subset of the available metrics and for each one supported, a sensor entity is available in HA. Some of these sensor entities will give erroneous readings whilst others give no readings at all. Sensor entities not supported by the charger will show as **Unknown** if you try to create a sensor entity for them. Below is a table of the metrics I've found useful for the Wallbox Pulsar Plus. Tables for other chargers will follow as contributions come in from owners of each supported charger.

## 3.4 Useful Entities for Wallbox Pulsar Plus

### 3.4.1 Metrics

- Energy Active Import Register or Energy Session (they give the same readings)
- Power Active Import (instantaneous charging power)
- Current Offered (maximum charging current available)
- Voltage (single phase models only, doesn't work on 3-phase)
- Frequency (single phase models only, doesn't work on 3-phase)
- Time Session (elapsed time from start of charging session)

### 3.4.2 Diagnostics

- **Status Connector** (shows the current state of available/preparing/charging/finishing/suspended etc)
- **Stop Reason** (reason the charging session was stopped)

### 3.4.3 Controls

- **Charge Control**
- **Availability** (must be set to ON before EV is plugged in)
- **Maximum Current** (sets maximum charging current available)
- **Reset**

## 3.5 Useful Entities for EVBox Elvi

### 3.5.1 Metrics

- **Current Offered** (maximum charging current available)
- **Time Session** (elapsed time from start of charging session)
- **Temperature** (internal charger temperature)

### 3.5.2 Diagnostics

- **Status Connector** (shows the current state of available/preparing/charging/finishing/suspended etc)
- **Stop Reason** (reason the charging session was stopped)

### 3.5.3 Controls

- **Charge Control**
- **Availability** (OFF when something causes a problem or during a reboot etc)
- **Maximum Current** (sets maximum charging current available)
- **Reset**

## 3.6 Useful Entities and Workarounds for United Chargers Grizzl-E

Comments below relate to Grizzl-E firmware version 5.633, tested Oct-Nov 2022.

### 3.6.1 Metrics

The Grizzl-E updates these metrics every 30s during charging sessions:

- **Current Import** (current flowing into EV)
- **Power Active Import** (power flowing into EV)
- **Energy Active Import Register** (cumulative energy supplied to EV during charging session. Resets to zero at start of each session)
- **Time Session** (elapsed time from start of charging session)

### 3.6.2 Diagnostics

- **Status Connector** (current charger state: available/preparing/charging/finishing/suspended etc)
- **Stop Reason** (reason the charging session was stopped)
- **Latency Pong** (elapsed time for charger's response to internet ping. Good for diagnosing connectivity issues. Usually less than 1000ms)
- **Version Firmware** (charger firmware version and build)

### 3.6.3 Controls

- **Charge Control** (User switches to ON to start charging session, once charger is in Preparing state. Can be automated in HA - see this [comment in Issue #442](#) for details)
- **Availability** (ON when charger is idle. OFF during active charging session, or when something causes a problem)
- **Maximum Current** (sets maximum charging current available. Reverts to value set by charger's internal DIP switch following reboots; tweak slider to reload)

## 3.7 Useful Entities for Vestel EVC-04 Wallboxes

### 3.7.1 Metrics

- **Energy Active Import Register** (cumulative energy supplied to EV during charging session. Resets to zero at start of each session)
- **Energy Active Import Interval** (in case you need the energy spent in total for the current charging session)
- **Power Active Import** (instantaneous charging power)
- **Current Import**
- **Time Session** (elapsed time from start of charging session)

### 3.7.2 Diagnostics

- **Status Connector** (shows the current state of available/preparing/charging/finishing/suspended etc)
- **Stop Reason** (reason the charging session was stopped)

### 3.7.3 Controls

- **Charge Control**
- **Availability** (must be set to ON before EV is plugged in)
- **Maximum Current** (sets maximum charging current available)
- **Reset**

### 3.7.4 OCPP Compatibility Issues

Grizzl-E firmware has a few OCPP-compliance defects, including responding to certain OCPP server messages with invalid JSON. Symptoms of this problem include repeated reboots of the charger. By editing the OCPP server source code, one can avoid these problematic messages and obtain useful charger behaviour. ChargeLabs (the company working on the Grizzl-E firmware) expects to release version 6 of the firmware in early 2023, which may fix these problems.

The workaround consists of:

- checking the *Skip OCPP schema validation* checkbox during OCPP server configuration
- commenting-out several lines in `/config/custom_components/ocpp/api.py` and adding a few default values to the OCPP server source code. Details are in this [comment in Issue #442](#)

**SUPPORT**

If you need help, check out our [forum](#) or submit an [issue](#).





## DEBUGGING

To enable debug logging for this integration and related libraries you need to update your Home Assistant `configuration.yaml` file:

```
logger:
  default: info
  logs:
    custom_components.ocpp: debug
```

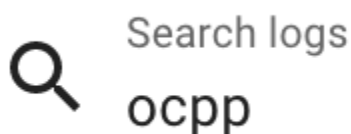
See [Home Assistant Logger](#) for more info.

After a restart detailed log entries will appear in `/config/home-assistant.log`. The log file can be displayed in your webbrowser, by selecting:

Configuration / Settings / Logs / LOAD FULL HOME ASSISTANT LOG



You can filter for OCPP related messages by typing 'ocpp' in the 'search logs' box at the top of the page.



A typical log for a working connection should look like this:

```
2022-03-16 16:33:08 INFO (MainThread) [custom_components.ocpp] {'host': '0.0.0.0', 'port': 9000, 'csid': 'central', 'cpid': 'pulsar', 'meter_interval': 60, 'idle_interval': 900, 'websocket_close_timeout': 10, 'WEBSOCKET_PING_TRIES': 2, 'websocket_ping_interval': 20, 'websocket_ping_timeout': 20, 'skip_schema_validation': False, 'monitored_variables': 'Energy.Active.Import.Register,Energy.Reactive.Import.Register,Energy.Active.Import.Interval,Energy.Reactive.Import.Interval,Power.Active.Import,Power.Reactive.Import,Power.Offered,Power.Factor,Current.Import,Current.Offered,Voltage,Frequency,RPM,SoC,Temperature,Current.Export,Energy.Active.Export.Register,Energy.Reactive.Export.Register,Energy.Active.Export.Interval,Energy.Reactive.Export.Interval,
```

(continues on next page)

(continued from previous page)

```

↪Power.Active.Export,Power.Reactive.Export'}
2022-03-16 16:35:40 INFO (MainThread) [custom_components.ocpp] Websocket Subprotocol
↪matched: ocpp1.6
2022-03-16 16:35:40 INFO (MainThread) [custom_components.ocpp] Charger websocket path=/
↪pulsar
2022-03-16 16:35:40 INFO (MainThread) [custom_components.ocpp] Charger pulsar connected
↪to 0.0.0.0:9000.
2022-03-16 16:35:40 DEBUG (MainThread) [custom_components.ocpp] Received boot
↪notification for pulsar: {'charge_point_serial_number': '88034', 'charge_point_vendor
↪': 'Wall Box Chargers', 'meter_type': 'Internal NON compliant', 'meter_serial_number':
↪', 'charge_point_model': 'PLP1-0-2-4', 'iccid': '', 'charge_box_serial_number': '88034
↪', 'firmware_version': '5.5.10', 'imsi': ''}
2022-03-16 16:35:40 DEBUG (MainThread) [custom_components.ocpp] Updating device info
↪pulsar: {'charge_point_serial_number': '88034', 'charge_point_vendor': 'Wall Box
↪Chargers', 'meter_type': 'Internal NON compliant', 'meter_serial_number': '', 'charge
↪point_model': 'PLP1-0-2-4', 'iccid': '', 'charge_box_serial_number': '88034',
↪'firmware_version': '5.5.10', 'imsi': ''}
2022-03-16 16:35:42 INFO (MainThread) [custom_components.ocpp] Supported feature
↪profiles: Core,FirmwareManagement,LocalAuthListManagement,SmartCharging,RemoteTrigger
2022-03-16 16:35:42 INFO (MainThread) [custom_components.ocpp] Supported feature
↪profiles: Core,FirmwareManagement,LocalAuthListManagement,SmartCharging,RemoteTrigger
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] Get Configuration for
↪NumberOfConnectors: 1
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] Get Configuration for
↪NumberOfConnectors: 1
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] Get Configuration for
↪HeartbeatInterval: 3600
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] Get Configuration for
↪HeartbeatInterval: 3600
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] 'pulsar' post connection
↪setup completed successfully
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] trigger status
↪notification for connector=0
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] 'pulsar' post connection
↪setup completed successfully
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] trigger status
↪notification for connector=0
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] trigger status
↪notification for connector=1
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] trigger status
↪notification for connector=1
2022-03-16 16:36:00 DEBUG (MainThread) [custom_components.ocpp] Connection latency from
↪'central' to 'pulsar': ping=2.0 ms, pong=13.0 ms
2022-03-16 16:36:20 DEBUG (MainThread) [custom_components.ocpp] Connection latency from
↪'central' to 'pulsar': ping=2.0 ms, pong=9.0 ms

```

To debug issues with establishing the ocpp connection, you can enable debug logging for websockets.server:

```

logger:
  default: info
  logs:
    websockets.server: debug

```

Filtering for websockets.server should yield something like this:

```

2022-03-16 16:33:08 INFO (MainThread) [websockets.server] server listening on 0.0.0.
↳0:9000
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] = connection is CONNECTING
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < GET /pulsar HTTP/1.1
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < Connection: Upgrade
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < Host: homeassistant.fritz.
↳box:9000
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < Sec-WebSocket-Key:
↳VLpFdctBQgYB6ZokyO2m3Q==
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < Sec-WebSocket-Protocol:
↳ocpp1.6
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < Sec-WebSocket-Version: 13
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < Upgrade: websocket
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < User-Agent: WebSocket++/0.8.
↳2
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > HTTP/1.1 101 Switching
↳Protocols
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > Upgrade: websocket
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > Connection: Upgrade
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > Sec-WebSocket-Accept:
↳hLE0rT2uOtRgVH4VLWoK8K7McNU=
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > Sec-WebSocket-Protocol:
↳ocpp1.6
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > Date: Wed, 16 Mar 2022
↳15:35:40 GMT
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > Server: Python/3.9
↳websockets/10.2
2022-03-16 16:35:40 INFO (MainThread) [websockets.server] connection open
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] = connection is OPEN
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < TEXT '[2,"4a7920fe-1ded-
↳48ff-b9c8-ff8f33bc8118","Boot...: "5.5.10","imsi": ""}]' [318 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > TEXT '[3,"4a7920fe-1ded-
↳48ff-b9c8-ff8f33bc8118",{"cur...0,"status":"Accepted"}]' [129 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < TEXT '[2,"336a0acf-3117-
↳4e72-99c6-f4ae31acb131","Stat...2022-03-16T15:35:40Z"]]' [211 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > TEXT '[3,"336a0acf-3117-
↳4e72-99c6-f4ae31acb131",{}]' [45 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < TEXT '[2,"654f6701-639c-
↳4398-8608-a0c7d8287465","Stat...2022-03-16T15:35:40Z"]]' [211 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > TEXT '[3,"654f6701-639c-
↳4398-8608-a0c7d8287465",{}]' [45 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < TEXT '[2,"694f0dac-fad4-
↳44e6-891c-23d535674cfd","Mete... 0,"transactionId": 0}]' [304 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > TEXT '[3,"694f0dac-fad4-
↳44e6-891c-23d535674cfd",{}]' [45 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < TEXT '[2,"c2c18e7a-b6fc-
↳40e4-ba5d-0423bf68d23d","Mete... 1,"transactionId": 0}]' [304 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > TEXT '[3,"c2c18e7a-b6fc-
↳40e4-ba5d-0423bf68d23d",{}]' [45 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < TEXT '[2,"5191e2e7-f555-
↳48b3-8b08-626679df5a80","Mete... 0,"transactionId": 0}]' [304 bytes]

```

(continues on next page)

(continued from previous page)

```
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > TEXT '[3,"5191e2e7-f555-  
↪48b3-8b08-626679df5a80",{ }]' [45 bytes]
```

## DEVELOPMENT

It is recommended to use Visual Studio Code, and run home assistant in a devcontainer. See <https://hacs.xyz/docs/developer/devcontainer>

Online development is supported through [GitHub Codespaces](#)