

# OCPP connection configuration

Version 1.4, 15-07-2024  
[internal & external usage]

**ecotap**<sup>®</sup>  
A brand of  **legrand**<sup>®</sup>

# Version History

Version	Date	Author
1.0	27-07-2023	CTO
1.1	23-08-2023	CTO
1.2	24-08-2023	CTO
1.3	7-03-2024	Product Owner (R&D)
1.4	15-7-2024	Product Owner (R&D)
1.4	17-7-2024	CTO

## History of changes:

- Version 1.0:
  - Creation
- Version 1.1:
  - Correct the com\_Endpoint format
- Version 1.2:
  - Rewording
- Version 1.3:
  - Added section about the use of ECCLite application
- Version 1.4:
  - Added note about OCPPInfo field.
  - Added limitation related to configuration (OCPPID).

# Type of controller

Ecotap has different type of controllers inside the chargers. Currently we have 3 types of controllers which has their own specific way of configuration for setting up the connectivity.

The 3 types of controllers are, Ecotap name:

- CPA controller
- EVC 4.X controller
- EVC 5.X controller

CPA controller:

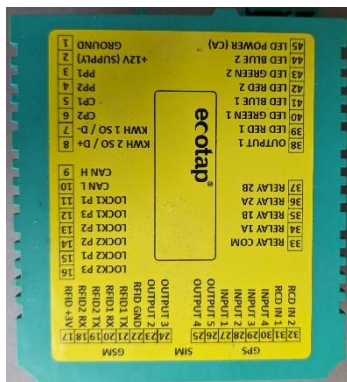


Image 1: CPA controller TOP view



Image 2: CPA controller SIDE view

EVC 4.X controller:

The controller Serial Number start always with:

- 117XXXX



Image 3: EVC 4.X controller TOP view

EVC 5.X controller:

The controller Serial Number start always with:

- 181XXXX for AC option
- 185XXXX for AC+ option
- 189XXXX for DC option



Image 4: EVC 5.X controller TOP view

# OCPP connection configuration

## A. CPA controller

The OCPP connection of the CPA controller can be done via OCPP by using the ChangeConfiguration.req function. The settings in the CPA controller for the endpoint URL is split up in smaller configuration items. These are the following configuration keys:

- **BackOffice-URL**
- **BackOffice-Portnr**
- **BackOffice-Path**
- **OCPPID**

Below two examples of an endpoint URL one unsecured which is prepended with "ws://" and one secured that is prepended with "wss://".

The **BackOffice-URL** does not have to be a domain name, an IP address can also be used.

### *For Unsecured websocket connection*

For example, if the full URL to connect to should be

"ws://www.example.com:80/ocpp1.6/path/TESTOCPPID" the settings should be as follows:

#### **Backoffice-URL**

"ws://www.example.com"

#### **Backoffice-Portnr**

"80"

#### **Backoffice-Path**

"/ocpp1.6/path/"

#### **OCPPID**

"TEST-OCPPID"

### *For Secured websocket*

For example, if the full URL is as follows "wss://192.168.255.8:443/ocpp1.6/secured/TEST-OCPPID" the settings should be as follows:

#### **Backoffice-URL**

"wss://192.168.255.8"

#### **Backoffice-Portnr**

"443"

#### **Backoffice-Path**

"/ocpp1.6/secured/"

#### **OCPPID**

"TEST-OCPPID"

Additional information:

If you want to change the **AuthorizationKey** you can send a ChangeConfiguration.req. The configuration key is write only which means you can only write new **AuthorizationKey** to the controller but not read out due to security reasons.

The key format is this one: <password>, the username will use the parameter **OCPPID**, here an example of **AuthorizationKey**: "9N8gGyS8Un7g4IY9dRIC".

So, the Sec-WebSocket-Key inside the header will be: "TEST-OCPPID:9N8gGyS8Un7g4IY9dRIC".

## B. EVC 4.X controller

The OCPP connection of the EVC 4.X controller can be done via OCPP by using the ChangeConfiguration.req function. The setting of in the EVC 4.X controller for the endpoint URL is split up in smaller configuration items. There are following configuration keys:

- **com\_Endpoint**
- **com\_OCPCPID**
- **com\_ProtType**
- **com\_Options**

*For Unsecured websocket connection*

For an unsecured websocket "ws://", the **useTLS** of **com\_Options** must be put to 0.

For example, if the full URL to connect to should be

"ws://www.example.com:80/ocpp1.6/path/TESTOCPCPID" the settings should be as follows:

**com\_Endpoint**

"www.example.com:80/ocpp1.6/path/#OSN#" (#OSN# will be replace by com\_OCPCPID)

**com\_OCPCPID**

"TEST-OCPCPID" (when this configuration is changed, the charger sill restart after 60 seconds)

(Maximum length = 25 characters)

**com\_ProtType**

"OCPP1.6J"

**com\_Options**

"Events=1,BlockBeforeBoot=1,Wdt=0,updSendInIdle=0blockLgFull=0,useTLS=0,conMaster=0"

*For Secured websocket*

The EVC 4.X does not support the secured websocket connection.

## C. EVC 5.X controller

The OCPP connection of the EVC 5.X controller can be done via OCPP by the ChangeConfiguration.req function. The setting of in the EVC 5.X controller for the endpoint URL is split up in smaller configuration items. There are following configuration keys:

- **com\_Endpoint**
- **com\_OCPPID**
- **com\_ProtType**
- **com\_Options**

### *For Unsecured websocket connection*

For a non-secured websocket “ws://”, the **useTLS** of **com\_Options** must be put to 0.

For example, if the full URL to connect to should be

“ws://www.example.com:80/ocpp1.6/path/TESTOCPPID” the settings should be as follows:

#### **com\_Endpoint**

“www.example.com:80/ocpp1.6/path/#OSN#” (#OSN# will be replace by com\_OCPPID)

#### **com\_OCPPID**

“TEST-OCPPID” (when this configuration is changed, the charger sill restart after 60 seconds)

(Maximum length = 25 characters)

#### **com\_ProtType**

“OCPP1.6J”

#### **com\_Options**

“Events=1,BlockBeforeBoot=1,Wdt=0,updSendInIdle=0blockLgFull=0,useTLS=0,conMaster=0”

### *For Secured websocket*

For a secured websocket “wss://”, the **useTLS** of **com\_Options** must be put to 1.

For example, if the full URL is as follows “wss://192.168.255.8:443/ocpp1.6/secured/TEST-OCPPID” the settings will be as follows:

#### **com\_Endpoint**

“192.168.255.8:443:80/ocpp1.6/secured/#OSN#” (#OSN# will be replace by com\_OCPPID)

#### **com\_OCPPID**

“TEST-OCPPID” (when this configuration is changed, the charger sill restart after 60 seconds)

#### **com\_ProtType**

“OCPP1.6J”

#### **com\_Options**

“Events=1,BlockBeforeBoot=1,Wdt=0,updSendInIdle=0blockLgFull=0,useTLS=1,conMaster=0”

### Additional information:

If you want to change the **AuthorizationKey** you can send a ChangeConfiguration.req. The configuration key is write only which means you can only write new AuthorizationKey to the controller but not read out due to security reasons.

The key format is this one: <username>:<password>, where the username should be the

**com\_OCPPID**, here an example of **AuthorizationKey**:

“TEST-OCPPID:9N8gGyS8Un7g4IY9dRIC”

# Application for connection configuration

## ECCLite

ECCLite is an application dedicated for owners, installers and operators of charging stations. Everything that can be done on this software tool, must in principle be done via the OCPP backend platforms. As the Ecotap stations are made for convenient remote control, in batch. That is also the case for all parameters needed to determine the power and grid settings that match your charging infrastructure.

Because Ecotap Charging Stations are infrastructure objects, the OCPP connectivity to the selected backend platform is pre-configured in the factory. If connectivity is lost or connectivity settings are accidentally wiped and/or contracts with the backend provider are terminated and a switch to a new party is needed. You will need to reconfigure connectivity yourself.

To connect an OCPP backend platform, you will need receive information from the platform provider. Namely, the link to the backend. Called an Endpoint.

In most cases it will look like this:

### Endpoint URL:

*“wss://devices.ecotap.com/registry/ocpp/NL\*ECO\*1000”*

The [ NL\*ECO\*1000 ] part is unique to a singular charging station and it’s backend page, called the OCPP-ID. Sometimes, if the backend has a kind of security lair. You receive one Token per charging station as well. That will match it’s unique charging station OCPP-ID. It will look like here below;

**Token:** *“53Umkk1q7rEM”*

The above information for the Endpoint and OCPPID will be split in the following fields.

<input type="checkbox"/> authorizationKey	NL*ECO*1000:53Umkk1q7rEM
<input type="checkbox"/> com_Endpoint	devices.ecotap.com:443/registry/ocpp/#OSN#
<input type="checkbox"/> com_OCPCID	NL*ECO*1000
<input type="checkbox"/> com_Options	comMaster=0,Events=1,BlockBeforeBoot=1,Wdt=0,updSendIdle=0,UseTLS=1,blockLgFull=0

In this case the [ wss:// ] in the endpoint link you receive from the CPO is removed.

If the link was [ wss:// ] you place in [ com\_Options ] the value UseTLS=1.

If the link was [ ws:// ] you place in [ com\_Options ] the value UseTLS=0.

As you can see after the [ .com ] part, a port number is added.

- Port :80 is WS connection.
- Port :443 is WSS connection.

The [ NL\*ECO\*1000 ] part is replaced by [ #OSN# ], that will mean that now the endpoint to this backend is not anymore unique per charger, but is applicable to every charging station connected to this backend.

The unique OCPPID is then filled in after [ com\_OCPPID ]. And this is the parameter unique to every charging station.

If in the cases this charging station and OCPPID need an [ authorizationKey ] you will add that after the parameter. In that value field you start with the OCPPID, and [ : ] an after that the per charger unique key. In this example after [ authorizationKey ] it will look like this;

[ NL\*ECO\*1000:53Umkk1q7rEM ].

Mind that you can set this parameter and after that you can't read it again. This is for safety.

#### **OCPP Info:**

Mind that, there is an additional parameter that needs to be completed for a connection to work:

**com OCPPInfo**

modelName=DC60,vendorname=Ecotap,CpSn=1891351

In each standard factory setting the charger receives its modelName, its vendorname=Ecotap and its CpSn (ChargePointSerialNumber).

If by some cause this info field is wiped, the connection to the backend will be hampered. Thus you need to set back to the correct info.