PSPowEr.Pl ul. Szafirowa 56, 62-023 Kamionki, tel. 502-355-433, email: biuro@pspower.pl

PSP EVSE v1.0 Controller

WebAPI Programming Guide

Table of Contents

Change log

Change w PG:

- v1.0 – First release.
- V1.1 – ActualDurationTotalSeconds in Status
- …

Introduction

PSP EVSE v1.0 controller have data exchange interface based on WebAPI in JSON format.

Status

Url:	http://<Host>/webapi/v1/Status
Access method:	GET
Access type:	Read only

Objects in response:

Object	Type	Description
TimeStamp	Number	Time stamp [ms] from system starts.
SystemVersion	String	System version [Major.Minor.Build]
WiFiInfo	String	Actual WiFi network [SSID:RSSI[%]]

EvseState	Number	EVSE state according to enumeration:
		typedef enum {//cpEvseStateEnum_t
		EVSE_WAITING = 0,
		EVSE_IDLE_A1_A2,
		EVSE_RELAY_ERR_F,
		EVSE_EPO_F,
		EVSE_EV_CONNECTED_B1_B2,
		EVSE_EV_CHARGE_C1_C2,
		EVSE_EV_VENT_D1_D2,
		EVSE_EV_DIODE_ERR,
		EVSE_EV_ERR_E,
		EVSE_EV_GENERAL_ERR,
		EVSE_PP_ERR_F,
		EVSE_LOCK_ERR_F
		}cpEvseStateEnum_t;

ThreePhaseCharging	Boolean	true – three phase charging, false – one phase charging
ActualCurrent	Number	Actual charging current [A]
PPCodedCurrent	Number	Only in mode C case B. Current limit read from connected cable.
ActualPower	Number	Actual charging power [W]
ActualPowerAvailable	Number	Only when active limiting power consumption algorithm. Actual available power for charging [W]. Level depends form actual building power consumption and renewable power generation.
ActualEnergy	Number	Actual delivered energy [Ws].
ActualDuration	String	Session time [hh:mm:ss].
ActualDurationTotalSeconds	Number	Session time in seconds.
AutoStartAllowed	Boolean	Shows whether auto start is allowed.
StartStopAllowed	Boolean	Shows whether start/stop charging session is allowed.
BoostAllowed	Boolean	Shows whether BOOST function is allowed.
ChargerStarted	Boolean	Shows whether charging session is started.
ChargeBoosted	Boolean	Shows whether BOOST function is active.
Meter1Status	Number	Shows meter 1 (statistics) communication quality. Value > 0 – reading data errors gauge.
Meter2Status	Number	Shows meter 2 (control) communication quality. Value > 0 – reading data errors gauge.

Example: GET http://192.168.0.17/webapi/v1/Status
Response:


```
{
	"TimeStamp": 179760,
	"SystemVersion": "0.9.1",
	"WiFiInfo": "559EVSE:60%",
	"EvseState": 4,
	"ThreePhaseCharging": true,
	"ActualCurrent": 16,
	"PPCodedCurrent": 20,
	"ActualPower": 11040,
	"ActualPowerAvailable": 22080,
	"ActualEnergy": 0,
	"ActualDuration": "00h:00m:00s",
	"ActualDurationTotalSeconds": 3774,
	"AutoStartAllowed": false,
	"StartStopAllowed": true,
	"BoostAllowed": true,
	"ChargerStarted": true,
	"ChargeBoosted": false,
	"Meter1Status": 0,
	"Meter2Status": 0
}
```


EVSE Control

Url:	http://<Host>/webapi/v1/EvseCtrl	
Access method:	POST	
Access type:	Write only	
Allowed objects:		
Object	Type	Description
ChargeStartToggle	Boolean	true = toggle charging START/STOP.
ChargeBoostToggle	Boolean	true = toggle BOOST function.
EVSEPowerAvailable	Number	Write actual available power for charging. Value came from external home automation system. [W]

Example: POST http://192.168.0.17/webapi/v1/EvseCtrl


```
{
	"ChargeStartToggle": true,
	"ChargeBoostToggle" : false,
	"EVSEPowerAvailable": 0
}
```


Response:


```
{
	"ChargeStartToggle": "Set=>False",
	"ChargeBoostToggle": "no action",
	"EVSEPowerAvailable": "supervisor IP not set"
}
```

Energy meter - statistics

Url:	http://<Host>/webapi/v1/Meter1
Access method:	GET
Access type:	Read only

Returning container object „Meter 1”, which include time stamp and sub container „Modbus Data” which include objects with meter data:

```
{
        Meter1:{
                        TimeStamp: <X>,
                        ModbusData:{
                                                FCxADDRESS:{ <Data>},
                                        }
                }
}
```

Each object include:

Object	Type	Description
Desc	String	Meter register description according to meter manufacturer documentation.
ErrCounter	Number	Register reading - error counter
Value	Number	Register value
DataStatus	String	Validity Status: Valid – good. Not available – obsolete or not available.

Example: GET http://192.168.0.17/webapi/v1/Meter1
Response:


```
{
        "Meter1": {
        "TimeStamp": 1826814,
        "ModbusData": {
                                "0x04_0x0034": {
                                "Desc": "TotalSystemPower[W]",
                                "ErrCounter": 0,
                                "Value": 475.2041931152344,
                                "DataStatus": "Valid"
                                },
                                "0x04_0x0048": {
                                "Desc": "TotalImport[kWh]",
                                "ErrCounter": 0,
                                "Value": 0.3310000002384186,
                                "DataStatus": "Valid"
                                },
                                "0x04_0x004a": {
                                "Desc": "TotalExport[kWh]",
                                "ErrCounter": 0,
                                "Value": 0,
                                "DataStatus": "Valid"
                                },
                                "0x04_0x0156": {
                                "Desc": "Total[kWh]",
                                "ErrCounter": 0,
                                "Value": 0.3310000002384186,
                                "DataStatus": "Valid"
```

```
                        },
                        "0x04_0x0180": {
                        "Desc": "ResettableTotalActiveEnergy[kWh]",
                        "ErrCounter": 0,
                        "Value": 0.3310000002384186,
                        "DataStatus": "Valid"
                        },
                        "0x04_0x0184": {
                        "Desc": "ResettableImportActiveEnergy[kWh]",
                        "ErrCounter": 0,
                        "Value": 0.3310000002384186,
                        "DataStatus": "Valid"
                        },
                        "0x04_0x0186": {
                        "Desc": "ResettableExportActiveEnergy[kWh]",
                        "ErrCounter": 0,
                        "Value": 0,
                        "DataStatus": "Valid"
                        },
                        "0x04_0x0500": {
                        "Desc": "TotalImportActivePower[W]",
                        "ErrCounter": 0,
                        "Value": 476.0400695800781,
                        "DataStatus": "Valid"
                        },
                        "0x04_0x0502": {
                        "Desc": "TotalExportActivePower[W]",
                        "ErrCounter": 0,
                        "Value": 0,
                        "DataStatus": "Valid"
                        }
                    }
        }
}
```


Energy meter - control

Url:	http://<Host>/webapi/v1/Meter2
Access method:	GET
Access type:	Read only
Data structure according to statistics meter.	

System configuration

Url:	http://<Host>/webapi/v1/SystemConfig	
Access method:	GET	
Access type:	Read only	
Objects in response:		
Object	Type	Description
TimeStamp	Number	Time stamp [ms] from system starts.
SystemVersion	String	System version [Major.Minor.Build]
ChargerMode	Number	Mode/Case: 0 - Mode 3 Case C, 1 - Mode 3 Case B
LockMethod	Number	Cable plug locking method. Refers to case B: 0 – Solenoid type lock, 1 – Servomotor type lock
ChargeCurrent	Number	Maximum charging current.
ChargeCurrentBoost	Number	Maximum charging current – BOOST function.
Meter1Type	Number	Statistics energy meter type: 0 – None, 1 – Modbus SDM72D-M, 2 – Modbus NMID30-2, 3 – Modbus SDM120M, 4 – IP SDM72D-M, 5 – IP NMID30-2, 6 – IP SDM120M
Meter1Ip	String	Statistics energy meter IP address (bridge MipB).
Meter2Type	Number	Control energy meter type: 0 – None, 1 – Modbus SDM72D-M, 2 – Modbus NMID30-2, 3 – Modbus SDM120M, 4 – IP SDM72D-M, 5 – IP NMID30-2, 6 – IP SDM120M, 7 – Zdalna kontrola
Meter2Ip	String	Control energy meter IP address (bridge MipB) or IP address external supervisor if set.
PhaseCurrentLimit	Number	Phase current limit of control energy meters if it delivers current for individual phases.
ControlStrategy	Number	Limiting power consumption algorithm: 0 – disabled, 1 – enabled – limit power
ControlLimit	Number	Power limit for limiting power algorithm.
ChargeAutoStart	Boolean	Allowing auto start charging session: true – allowed, false – not allowed
Charge3Phase	Boolean	Three phase charging: true – three phase, false – one phase

Example: GET http://192.168.0.17/webapi/v1/SystemConfig
Response:


```
{
        "TimeStamp": 1661687,
        "SystemVersion": "0.5.0",
        "ChargerMode": 0,
        "LockMethod": 0,
        "ChargeCurrent": 6,
        "ChargeCurrentBoost": 32,
        "Meter1Type": 1,
        "Meter1Ip": "192.168.0.30",
        "Meter2Type": 2,
        "Meter2Ip": "192.168.0.30",
        "PhaseCurrentLimit": 25,
        "ControlStrategy": 1,
        "ControlLimit": 0,
        "ChargeAutoStart": true,
        "Charge3Phase": true
}
```


Url:	http://<Host>/webapi/v1/SystemConfig	
Access method:	POST	
Access type:	Write only	
Allowable objects:		
Object	Type	Description
ChargerMode	Number	According to GET
LockMethod	Number	
ChargeCurrent	Number	
ChargeCurrentBoost	Number	
Meter1Type	Number	
Meter1Ip	String	
Meter2Type	Number	
Meter2Ip	String	
PhaseCurrentLimit	Number	
ControlStrategy	Number	
ControlLimit	Number	
ChargeAutoStart	Boolean	
Charge3Phase	Boolean	

Setting the energy meter to IP type is possible only with setting the correct IP address simultaneously.

System information

Url:	http://<Host>/webapi/v1/SystemInfo	
Access method:	GET	
Access type:	Read only	
Objects in response:		
Object	Type	Description
TimeStamp	Number	Time stamp [ms] from system starts.
SystemVersion	String	System version [Major.Minor.Build]
ProductType	String	Product type signature.
MAC	String	Mac address.
WiFiInfo	String	Information about WiFi: SSID,RSSI dB/%
HeapInfo	String	Heap allocation information
GeneralInfo	String	General system condition.
OTAMsg	String	OTA system message.
OTACheckAllowed	Boolean	Shows whether OTA check new firmware function is allowed.
OTAUpdateAllowed	Boolean	Shows whether OTA update firmware function is allowed.

Example: GET http://192.168.0.17/webapi/v1/SystemInfo
Response:


```
{
        "TimeStamp": 520144,
        "SystemVersion": "0.9.2",
        "ProductType": "01100200",
        "MAC": "E0:E2:E6:52:AA:D4",
        "WiFiInfo": "559EVSE,RSSI:-49dB/68%",
        "HeapInfo": "0x022DF0/0x02D230",
        "GeneralInfo": "System: Failure free",
        "OTAMsg": "Operation not allowed in WiFi AP mode!",
        "OTACheckAllowed": true,
        "OTAUpdateAllowed": false
}
```


Url:	http://<Host>/webapi/v1/SystemInfo	
Access method:	POST	
Access type:	Write only	
Allowable objects:		
Object	Type	Description
SCode	String	Service code.
SCodeData	String	Data for service code.
UpdateCheckToggle	Boolean	It runs single check for new firmware available.
UpdateRunToggle	Boolean	It runs firmware update.

Service codes:

SCode	SCodeData	Description
Reset		System restart.
FactoryDefault		Sets factory default settings.
WiFiSetToAP		Sets WiFi to AP (when STA).

Example: POST http://192.168.0.17/webapi/v1/SystemInfo

```
{
	"Scode": "Reset",
	"SCodeData": ""
}
```

Response:

```
{
	"Reset": "Performing"
}
```