



Energy Automation Interface Specification

User Interface (UI)



TABLE OF CONTENTS

TABLE OF CONTENTS.....	1
1 SCOPE.....	2
2 USER INTERFACE FUNCTIONALITY	2
2.1 General System Status.....	2
3 MODES OF ENERGY MANAGEMENT	6
3.1 Energy Management in On-Grid Modes	6
3.2 Energy Management in Off-Grid Modes	9
4 RETURN CODES.....	15



1 SCOPE

The scope of this document is to describe the functionality of the User Interface (UI) to be included in Energy Automation Systems (EA Systems) to properly interact with sonnen's ecoLinx product line (ecoLinx 10 through 20). The Home Automation System would communicate with the ecoLinx system using sonnen's API.

2 USER INTERFACE FUNCTIONALITY

The UI for the ecoLinx product line is intended to provide accessibility to:

2.1 GENERAL SYSTEM STATUS

Information relevant to the state of operation of the ecoLinx.

- 2.1.1 **Command to retrieve system status:** The below command should be used to get system status data for different variables as described in **Table 2-1** below,

Local API command for system status

Curl -v -X GET http://DEVICE_IP:8080/api/v1/status

Where,

- DEVICE_IP – The Local IP Address of the Device.

Server API command for system status

curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/v1/status" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"

Where,

- SERIAL – Serial number of the system to connect
- TOKEN – Bearer authentication Token

Expected Response

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 467

```
{"BackupBuffer": "100", "BatteryCharging": false, "BatteryDischarging": false, "Consumption_W": 0, "Fac": 60, "FlowConsumptionBattery": false, "FlowConsumptionGrid": false, "FlowConsumptionProduction": false, "FlowGridBattery": false, "FlowProductionBattery": false, "FlowProductionGrid": false, "GridFeedIn_W": 0, "IsSystemInstalled": 1, "OperatingMode": "10", "Pac_total_W": 0, "Production_W": 0, "RSOC": 100, "SystemStatus": "OnGrid", "Timestamp": "2019-05-03 08:54:49", "USOC": 100, "Uac": 248, "Ubat": 55}
```

Name of the Variable	Description
BackupBuffer	Backup buffer of the system
BatteryCharging	Boolean that indicates the charge status. True for charging
BatteryDischarging	Boolean that indicates the discharge status. True for discharging
Consumption_W	House consumption in watts
Fac	AC frequency in hertz
FlowConsumptionBattery	Boolean that indicates the energy flow at the installation site. True for battery feeds the consumption
FlowConsumptionGrid	Boolean that indicates the energy flow at the installation site. True for grid feeds the consumption
FlowConsumptionProduction	Boolean that indicates the energy flow at the installation site. True for production feeds the consumption
FlowGridBattery	Boolean that indicates the energy flow at the installation site. True for battery is charging from grid
FlowProductionBattery	Boolean that indicates the energy flow at the installation site. True for production is charging the battery
FlowProductionGrid	Boolean that indicates the energy flow at the installation site. True for production feeds into the grid
GridFeedIn_W	Grid Feed in negative is consumption and positive is feed in
IsSystemInstalled	System is installed or not
OperatingMode	Operating Mode of the system
Pac_total_W	Inverter AC Power greater than ZERO is discharging Inverter AC Power less than ZERO is charging
Production_W	PV production in watts
PowerMeter	Cummulative Energy Data collected by the production and consumption meters
RSOC	Relative state of charge
SystemStatus	Status of the system, "Ongrid" or "Offgrid"
Timestamp	System local time
USOC	User state of charge
Uac	AC voltage in volts
Ubat	Battery voltage in volts

Table 2-1: Description of Variables in Response & Return Codes



- 2.1.1.1 **State of the Grid:** Reflects whether the system is connected to the grid or operating in micro-grid mode (off-grid mode). Grid status can be determined by checking the value of the “SystemStatus” variable from **Table 2-1**,

if the value is “Ongrid” it means that the system is connected to Grid and if value is between “Offgrid” it means that the system is not connected to Grid.

- 2.1.1.2 **State of Power Flow:** Indicates the amount of power being generated, stored and consumed in the electrical system managed by the ecoLinX. This includes the amount of power fed into or taken from the grid and the production of PV (when installed). Power Flow can be determined by checking the value of “Consumption_W”, “Production_W” & “Pac_total_W” from **Table 2-1**.

Formulas for different States of Power Flow

Purchase from Grid = Consumption_W – [Production_W + Pac_total_W]

(If Purchase from Grid > 0, the ecoLinX is buying power from grid. Similarly, if Purchase from Grid < 0, the ecoLinX is feeding into the grid.)

Autonomy = (1 – [Energy Purchased / Energy Consumed]) *100

(This should be periodically calculated for the whole day. This needs to be collected for 7 days’ worth of time and Autonomy value updated as an average over those 7 days.)

- 2.1.1.3 **EA System connection:** Indicates proper connection between the ecoLinX and the EA System for remote monitoring and control. Developers can create a Heartbeat function in their system which checks for connection with the ecoLinX system.

- 2.1.1.4 **Status of Operation:** Indicates if there are any issues with the system in the form of errors or warning messages.

For example, these errors or warnings could be

- Irregular heartbeat – indicates a communication outage between EA System and ecoLinX.
- Unexpected or No Response to command(s) – if the EA System has sent commands to the ecoLinX for any operations and there is no response within 5-10 minutes, this error should be logged.

- 2.1.1.5 **Battery Modules:** The below command provides information on the number of Battery modules installed in the ecoLinX system.

Local API command for number of battery modules

Curl -v -X GET http://DEVICE_IP:8080/api/configuration/IC_BatteryModules



Server API command for number of battery modules

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/configuration/IC_BatteryModules" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where,

- IC_BatteryModules – Number of Installed Battery Modules

Using the above information, the user can calculate the total storage capacity of the ecoLinx system

Storage Capacity = IC_BatteryModules * 2 [kWh] (Each battery module has a capacity of 2 kWh)

2.1.1.6 **Inverter Max Capacity:** The below command provides the max power that the Inverter can output.

Local API command for max inverter power

Curl -v -X GET http://DEVICE_IP:8080/api/configuration/IC_InverterMaxPower_w

Server API command for max inverter power

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/configuration/IC_InverterMaxPower_w" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where,

- IC_InverterMaxPower_w – Max inverter power in Watts [W]

2.1.1.7 **Energy/Power Measurements:** The below command provides a group of power and energy measurements collected by the production and consumption meters.

Local API command for power meter data

Curl -v -X GET http://DEVICE_IP:8080/api/powermeter

Server API command for power meter data

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/powermeter" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where,

- a_lx – instantaneous current in amps associated with phase x
- channel – modbus channel identifier (sonnen internal)
- deviceid – modbus device identifier within specified channel (sonnen internal)
- direction – describing if this is the production or consumption meter
- error – 0-no error -1-comm error
- kwh_exported – amount of power fed back into the grid
- kwh_imported – amount of power received from the grid



- v_{lx_ly} – instantaneous voltage across phases x and y
- v_{lx_n} – instantaneous voltage from phase x to neutral
- va_total – instantaneous Total Volt-Amps measurement
- var_total – instantaneous Total Reactive Power measurement
- w_{lx} – instantaneous Real Power measurement on phase x
- w_total – instantaneous Total Real Power measurement

3 MODES OF ENERGY MANAGEMENT

For on-grid and off-grid conditions, there are several different energy management modes supported by the ecoLinX. The following subsections detail all the available energy management modes. The table below shows the various supported operating modes.

EM_OperatingMode	VALUE
Backup Mode	7
Self-Consumption Mode	8
Manual Mode	1
Time of Use Mode	10

Table 3-1: Description of operating modes and their values

3.1 ENERGY MANAGEMENT IN ON-GRID MODES

Below are the descriptions of all on-grid energy management modes with their associated API Commands to control the system.

- 3.1.1 **Backup Mode:** Intended to maintain an energy reserve for situations where the Grid is no longer available. During the off-grid period the energy would be dispensed to supply the demand of power from all the essential loads. Load management can be enabled to further extend the life of the batteries by the Developers.

- **Note:** if the ecoLinX is on-grid and set to Backup Mode, it will charge to and maintain 100% SOC on all batteries and only discharge power when the ecoLinX goes off-grid.

Local API command to switch to backup mode

```
curl -v -X GET http://DEVICE_IP:8080/api/setting?EM_OperatingMode=7
```

Server API command to switch to backup mode

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?EM_OperatingMode=7" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

- 3.1.2 **Self-Consumption Mode:** The ecoLinX monitors all energy sources (Grid, PV, Generator), loads, and Energy Reserve Percentage in order to minimize purchase of energy from the Grid.



Local API command to switch to self-consumption mode

```
Curl -v -X GET http://DEVICE_IP:8080/api/setting?EM_OperatingMode=8
```

Server API command to switch to self-consumption mode

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?EM_OperatingMode=8" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

- a. **Energy Reserve Percentage:** This setting allows the user to set a minimum State of Charge percentage (%SOC) to keep available for Backup purposes (i.e. a Zero %SOC setting will use all the battery capacity for Self-Consumption Mode).

Local API command to change value of energy reserve percentage

```
Curl -v -X GET http://DEVICE_IP:8080/api/setting?EM_USOC=VALUE
```

Server API command to change value of energy reserve percentage

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?EM_USOC=VALUE" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where,

- VALUE – Energy Reserve percentage (from 0 to 100 [%])

- 3.1.3 **Manual Mode:** This mode allows the user to manually charge or discharge the batteries. The user needs to provide the value for charging or discharging and based on the value, the ecoLinx system will charge until it reaches 100% or discharge until it reaches 0% User SOC (unless stopped by the user by changing the charge/discharge value to 0).

Local API commands to switch to manual mode, and set the setpoint for charge or discharge

```
Curl -v -X GET http://DEVICE_IP:8080/api/setting?EM_OperatingMode=1
```

```
Curl -v -X GET http://DEVICE_IP:8080/api/v1/setpoint/charge/VALUE
```

```
Curl -v -X GET http://DEVICE_IP:8080/api/v1/setpoint/discharge/VALUE (Not to be shown on User Interface)
```

Server API commands to switch to manual mode, and set the setpoint for charge or discharge

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?EM_OperatingMode=1" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/v1/setpoint/charge/VALUE" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```




```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/v1/setpoint/discharge/VALUE" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN" (Not to be shown on User Interface)
```

Where VALUE is,

- A power value from 0 to IC_InverterMaxPower_w. Here, the value of "IC_InverterMaxPower_w" can be determined by the following API commands:

Local API command for max inverter power

```
Curl -v -X GET http://DEVICE_IP:8080/api/configuration/IC_InverterMaxPower_w
```

Server API command for max inverter power

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/configuration/IC_InverterMaxPower_w" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

- 3.1.4 **Time of Use (TOU):** This mode allows users to set time windows where it is preferred to employ the use of stored energy (from PV) rather than consume from the grid. Below are brief descriptions for configuration parameters of TOU mode with their respective API commands.

Local API command to switch to Time of Use (TOU) mode

```
Curl -v -X GET http://DEVICE_IP:8080/api/setting?EM_OperatingMode=10
```

Server API command to switch to Time of Use (TOU) mode

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?EM_OperatingMode=10" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Single ToU:

Local API command to configure single low tariff time charge event

```
Curl -g -X GET  
http://DEVICE_IP:8080/api/setting?EM_ToU_Schedule=[{"start": "HH:MM", "stop": "HH:MM", "threshold_p_max": VALUE}]
```

Server API command to configure single low tariff time charge event

```
curl -g -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?EM_ToU_Schedule=[{"start": "HH:MM", "stop": "HH:MM", "threshold_p_max": VALUE}]" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```



Multiple ToU: To set multiple ToU windows, use the following API commands to set multiple low tariff time windows and charge events.

Local API command to configure multiple low tariff time charge event

```
Curl -g -X GET  
http://DEVICE_IP:8080/api/setting?EM_ToU_Schedule=[{%22start%22:%22HH:MM%22,%22stop%22:%22HH:MM%22,%22threshold_p_max%22:VALUE},{%22start%22:%22HH:MM%22,%22stop%22:%22HH:MM%22,%22threshold_p_max%22:VALUE}]
```

Server API command to configure multiple low tariff time charge event

```
curl -g -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?EM_ToU_Schedule=[{%22start%22:%22HH:MM%22,%22stop%22:%22HH:MM%22,%22threshold_p_max%22:VALUE},{%22start%22:%22HH:MM%22,%22stop%22:%22HH:MM%22,%22threshold_p_max%22:VALUE}]" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where,

- HH:MM is the desired time to be entered in equivalent **UTC** time in 24-hour time format.
- VALUE is power in Watts (it is the max allowed power from grid to charge the system during low tariff charge event)
 - Note: Each low tariff time window could use different VALUE (if needed)

Local API command to set the purchase power threshold limit during high tariff time

```
Curl -v -X GET http://DEVICE_IP:8080/api/setting?EM_US_HIGH_TARIFF_THRESHOLD=VALUE
```

Server API command to set the purchase power threshold limit during high tariff time

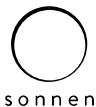
```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?/EM_US_HIGH_TARIFF_THRESHOLD=VALUE" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where,

- VALUE is Power in Watts (purchase power threshold during high tariff time window)

3.2 ENERGY MANAGEMENT IN OFF-GRID MODES

Below are the descriptions of all off-grid energy management modes with their associated API Commands to control the system.



- 3.2.1 **Micro grid Recovery at low SOC:** The ecoLinx will turn off the micro grid once the system has reached a Low RSOC state, which means that it will disconnect itself from PV as well. In order to reconnect with PV on the next day, the ecoLinx needs to turn on the micro grid to check if PV is available. This can be achieved by setting up three different time windows to check for PV. During this time window, the system will turn on the micro grid for 6 minutes to check for PV and if PV is available, the system will start charging. Else, it will again turn off the micro grid and wait for the next time window.

Local API command to enable micro grid recovery mode

```
Curl -v -X GET http://DEVICE_IP:8080/api/setting?EM_RE_ENABLE_MICROGRID=VALUE
```

Server API command to enable micro grid recovery mode

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?EM_RE_ENABLE_MICROGRID=VALUE" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where VALUE is,

- Equal to 1 to enable micro grid recovery during the below times
- Equal to 0 to disable micro grid recovery during the below times

Local API command to set micro grid recovery times one, two and three

```
Curl -v -X GET http://DEVICE_IP:8080/api/setting?EM_USER_INPUT_TIME_ONE=HH:MM
```

```
Curl -v -X GET http://DEVICE_IP:8080/api/setting?EM_USER_INPUT_TIME_TWO=HH:MM
```

```
Curl -v -X GET http://DEVICE_IP:8080/api/setting?EM_USER_INPUT_TIME_THREE=HH:MM
```

Server API command to set micro grid recovery times one, two and three

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?EM_USER_INPUT_TIME_ONE=HH:MM" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?EM_USER_INPUT_TIME_TWO=HH:MM" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?EM_USER_INPUT_TIME_THREE=HH:MM" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where,

- HH:MM is the desired time to be entered in equivalent **UTC** time in 24-hour time format.



- 3.2.2 **Charging Batteries using a Generator:** This mode is designed to charge batteries using a Generator while the system is in an off-grid condition. The ecoLinx supports both Manual & Automatic Generators for charging the batteries.

Local API command to read the value of Generator Type

```
Curl -v -X GET  
http://DEVICE_IP:8080/api/configuration/EM_US_GENRATOR_TYPE  
(This Command will provide the value of installed Generator type as a string: "automatic" or "manual")
```

Server API command to read the value of Generator Type

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/configuration/EM_US_GENRATOR_TYPE" -  
H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Local API command to change the Generator Type

```
Curl -v -X GET  
http://DEVICE_IP:8080/api/setting?EM_US_GENRATOR_TYPE=VALUE
```

Server API command to change the Generator Type

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?EM_US_GENRATOR_TYPE=VALUE" -  
-H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where VALUE is,

- Equal to "automatic" or "manual" with respect to actual generator type

Local API command to set the Generator charge power set point

```
Curl -v -X GET http://DEVICE_IP:8080/api/setting?EM_US_GEN_POWER_SET_POINT=VALUE
```

Server API command to set the Generator charge power set point

```
curl -X "GET" "https://core-  
api.sonnenbatterie.de/proxy/53456/api/setting?EM_US_GEN_POWER_SET_POINT=VALUE" -H "Accept:  
application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where VALUE is,

- Generator charge power set point in Watts

Local API command to configure automatic start of the Generator

(Note: this is only applicable for Automatic Generators)

```
Curl -v -X GET http://DEVICE_IP:8080/api/setting?EM_CHP_Min_SOC=VALUE
```



Server API command to configure automatic start of the Generator

(Note: this is only applicable for Automatic Generators)

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?EM_CHP_Min_SOC=VALUE" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where VALUE is,

- The minimum ecoLinx SOC at which point the Generator will turn on (generally it should be greater than the low SOC value and less than EM_CHP_Max_SOC defined below)

Local API command to identify the low SOC Value

```
Curl -v -X GET http://DEVICE_IP:8080/api/configuration/NVM_OffgridMinimalOperationSOC
```

Server API command to identify the low SOC Value

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/configuration/NVM_OffgridMinimalOperationSOC" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Local API command to identify EM CHP Max SOC

```
Curl -v -X GET http://DEVICE_IP:8080/api/configuration/EM_CHP_Max_SOC
```

Server API command to identify EM CHP Max SOC

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/configuration/EM_CHP_Max_SOC" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Local API command to configure automatic stop of the Generator

(Note: this is only applicable for Automatic Generators)

```
Curl -v -X GET http://DEVICE_IP:8080/api/setting?EM_CHP_Max_SOC=VALUE
```

Server API command to configure automatic stop of the Generator

(Note: this is only applicable for Automatic Generators)

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?EM_CHP_Max_SOC=VALUE" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where VALUE is,

- The maximum ecoLinx SOC at which point the Generator will turn off (generally it would be less than maximum SOC and greater than EM_CHP_Min_SOC)

Local API command to identify the EM CHP Min SOC



Curl -v -X GET http://DEVICE_IP:8080/api/configuration/EM_CHP_Min_SOC

Server API command to identify the EM_CHP_Min_SOC

curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/configuration/EM_CHP_Min_SOC" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"

Local API command to identify the maximum SOC

Curl -v -X GET http://DEVICE_IP:8080/api/configuration/EM_RSOCMax

Server API command to identify the maximum SOC

curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/configuration/EM_RSOCMax" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"

- 3.2.3 **Changing Frequency Shift Value:** Using this variable, the user can set custom frequency per the PV Inverter to disconnect with PV system at High RSOC. The default value is 609 (for 60.9 Hz).

Local API command to read the Frequency Shift value

(Not to be shown on User Interface)

Curl -v -X GET http://DEVICE_IP:8080/api/configuration/EM_FREQUENCY_SHIFT_VALUE

Server API command to read the Frequency Shift value

(Not to be shown on User Interface)

curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/configuration/EM_FREQUENCY_SHIFT_VALUE" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"

Where EM_US_FREQUENCY_SHIFT_VALUE VALUE returned is,

- The Frequency Shift value (Default value is 609 which corresponds to 60.9 Hz)

Local API command to set the Frequency Shift value

(Not to be shown on User Interface)

Curl -v -X GET http://DEVICE_IP:8080/api/setting?EM_FREQUENCY_SHIFT_VALUE =VALUE

Server API command to set the Frequency Shift value

(Not to be shown on User Interface)



```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?EM_FREQUENCY_SHIFT_VALUE=VALUE" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where VALUE,

- Has a format of XXX that corresponds to XX.X Hz. (range 609 to 649)

3.2.4 **Changing Value of Low RSOC & Very Low RSOC:** The user can change the values of Low and Very Low RSOC which are meant for turning off the micro grid and shutting down the system to prevent complete depletion of batteries, respectively.

Local API command to read the value of Low RSOC

(Not to be shown on User Interface)

```
Curl -v -X GET http://DEVICE_IP:8080/api/configuration/NVM_OffgridMinimalOperationSOC
```

Server API command to read the value of Low RSOC

(Not to be shown on User Interface)

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/configuration/NVM_OffgridMinimalOperationSOC" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where NVM_OffgridMinimalOperationSOC is,

- The value of Low RSOC in Percent (the default value is 10%)

Local API command to write the value of Low RSOC

(Not to be shown on User Interface)

```
Curl -v -X GET http://DEVICE_IP:8080/api/setting?NVM_OffgridMinimalOperationSOC=VALUE
```

Server API command to write the value of Low RSOC

(Not to be shown on User Interface)

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?NVM_OffgridMinimalOperationSOC=VALUE" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where VALUE is,

- The Low RSOC Value in percent (X-XXX [%]) at which a user's system will turn off the micro grid

Local API command to read the value of Very Low RSOC

(Not to be shown on User Interface)

```
Curl -v -X GET http://DEVICE_IP:8080/api/configuration/NVM_OffgridIslandingReEnableMinSOC
```



Server API command to read the value of Very Low RSOC

(Not to be shown on User Interface)

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/configuration/NVM_OffgridIslandingReEnableMinSOC" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where NVM_OffgridIslandingReEnableMinSOC is,

- The value of Very Low RSOC in Percent (the default value is 7%)

Local API command to write the value of Very Low RSOC

(Not to be shown on User Interface)

```
Curl -v -X GET http://DEVICE_IP:8080/api/setting?NVM_OffgridIslandingReEnableMinSOC=VALUE
```

Server API command to write the value of Very Low RSOC

(Not to be shown on User Interface)

```
curl -X "GET" "https://core-api.sonnenbatterie.de/proxy/SERIAL/api/setting?NVM_OffgridIslandingReEnableMinSOC=VALUE" -H "Accept: application/vnd.sonnenbatterie.api.core.v1+json" -H "Authorization: Bearer TOKEN"
```

Where VALUE is,

- The Very Low RSOC Value in percent (X-XXX [%]) at which a user's system will shut down.

4 RETURN CODES

Table 4-1 shows the return codes of a request response. This includes the local and the server API.

Return Code	Description
0	Request successfully received
1	Maximum number of retries executed
2	TCP connection ERROR
5	Invalid request path
8	Internal Database error
11	Invalid Setting
13	Internal Error
14	Restart of the system
16	Invalid HTTP method

Table 4-1: Description of Return codes