charge. fast. anywhere.

The flexible Wallbox.

Custom API

integrate not natively supported devices

Version 1.2.

1. Custom API

Web sever overview
This script implements a basic web server that listens for incoming connections and responds to GET requests with JSON data. The NRGkick communicates with this web server to retrieve device values and information.

Requirements
Please make sure that Python is installed on your system. The script uses the socket and json libraries, which are included in Python by default.

Configuration
Please fill the following parameters in Custom_API_Webserver.py with your device data:

It is possible to integrate currently unsupported devices or manufacturers via the Custom API.
To do this, the device data must be queried and communicated to the NRGkick via a web server. The Python script Custom_API_Webserver.py is provided by DiniTech.

1. deviceValues: In this JSON structure you define your device values. Inverters, smart meters, batteries and smart loads can be specified. You can include up to five different devices (with indices from 0-4). However, battery and smart loads can be added only one.

json

```
{
  "device_values": {
    "inverters": {
      "0": {
        "inv_power_ac_w": 2225,          #Set current ac inverter power in W
        "pv_energy_day_wh": 8750,        #Set day energy generation of the inverter in Wh
        "pv_energy_total_wh": 346890,    #Set total energy generation of the inverter in Wh
        "battery_connected": True        #Set if configured battery is connected to this inverter
      }
    },
    "energy_meters": {
      "0": {
        "grid_power_w": -542             #Set current grid power in W (-… feed-in, +… consumption)
      }
    },
    "batteries": {
      "0": {
        "bat_power_w": -749,             #Set current battery power in W (-… charge, +… discharge)
        "soc": 78.2,                     #Set current battery soc in %
        "mode": 0                        #Set current battery mode (0-normal)
      }
    },
    "smart_loads": {
      "0": {
        "sl_power_w": 1241,              #Set current smart load power in W
        "temp": 46.8,                    #Set current smart load temperature in °C
        "mode": 1                        #Set current smart load mode
      }
    },
    "load_power_w": 1683                 #Set current house load in W
  }
```

1. Custom API

```json
{
  "device_info": {
    "inverters": {
      "0": {
        "name": "Testinverter",        #Set name of inverter
        "model": "Testmodel",          #Set model of inverter
        "serial": "0123456789",        #Set serial number of inverter
        "max_dc_w": 5500,              #Set maximum / connected dc power in W
        "max_ac_w": 5000               #Set maximum ac power in W
      }
    },
    "energy_meters": {
      "0": {
        "name": "Testmeter",           #Set name of smart meter
        "model": "Testmodel",          #Set model of smart meter
        "serial": "0123456789",        #Set serial number of smart meter
        "location": 1                  #Set meter location
      }
    },
    "batteries": {
      "0": {
        "name": "Testbatterie",        #Set name of battery
        "model": "Testmodel",          #Set model of battery
        "serial": "0123456789",        #Set serial number of battery
        "capacity_wh": 4700.           #Set capacity of batter in Wh
      }
    },
    "smart_loads": {
      "0": {
        "name": "Testsmartload",       #Set name of smart load
        "model": "Testmodel",          #Set model of smart load
        "serial": "0123456789"         #Set serial number of smart load
      }
    }
  }
}
```

2. deviceInfo: In this JSON structure you define your device information. Inverters, smart meters, batteries and smart loads can be specified. You can provide up to five different devices (with indexes from 0-4) per category.
3. webserverHost: Set the IP address or hostname where the web server will run.
4. webserverPort: Specify the port on which the web server should listen for incoming connections. Choose an available port, for example, 1000.

Explanation of the script:

1. Import libraries: The script imports the required libraries socket for network communication and json for JSON processing.
2. Device values and device information: The script includes examples for the two variables deviceValues and deviceInfo. These JSON structures are communicated from the web server to the NRGkick with the data you define.
3. Socket configuration: the script creates a socket, binds it to the configured host and port, and makes it listen for incoming connections.
4. Handling connections: The script enters a loop to handle incoming client connections. For each connection:
    a. The request data is received and decoded from bytes into a string.
    b. The request headers are split to extract the request type (e.g. GET) and the path (e.g. /api/v1/values.json).
    c. The script responds based on the request type and path. It handles GET Requests for /api/v1/values.json and /api/v1/info.json.
    d. If the request path is not recognized, the script responds with a 404 error.
    e. If the request type is not GET, the script responds with a 500 error.

1. Custom API

5. Response Handling: the script creates an HTTP response with appropriate headers and the JSON data, which is interpreted by the NRGkick.
6. Close connection: After sending the response, the connection is closed.
7. Clean up socket: Once the loop is finished (e.g. by exiting the script), the server socket is closed.

Execute the scipt

1. Configure the deviceValues, deviceInfo, webserverHost and webserverPort variables according to your requirements.
2. Run the script using Python. Open a terminal or command prompt, navigate to the directory containing the script, and run the following command:

```
Terminal
python3 Custom_API_Webserver.py
```

3. The script starts the web server on the specified host and port. Subsequently, the status of the server is displayed.

2. Contact/Support

If you have any further questions or need help,
please feel free to contact us by phone or email:

DiniTech GmbH
+43 664 537 62 51
office@nrgkick.com

Support
+43 664 401 13 50
support@nrgkick.com