Unified-E WebHttp-JSON Adapter User Manual

Configure Web Endpoints and Datapoints

Software version 3.1.0.0, last updated: July 2025

Publisher: Unified-E AG, Winterthur, Switzerland

Content

1 General

With the help of the WebHttp-JSON adapter, datapoints managed by a web server can be accessed. This can be used, for example, to read data from already established devices or to create a bridge to the ERP system.

Two basic web interfaces are offered:

- Standard:
  Access is via standard calls, which are already supported by many devices that provide data via a web server. Additional web server programming is not required here.
- Unified-E (predefined):
  This requires programming of the REST interface for a web server. Extended functions are available compared to the "standard" web interface:
  - Authentication via username and password
  - Datapoint contexts: Here, the web server can return different values depending on the operator device name/user or depending on the HMI app language

Both web interfaces are described in the chapter 3 described in detail.

2 Configure General Parameters

The following general properties can be set with the WebHttp-JSON adapter.

Address:

The address of the web server (endpoint), such as the IP address, URL, or localhost.

Parameter settings:

- Web interface: Describes the way the endpoint adapter reads or writes datapoints from the web server:
  - Standard: Simple standard JSON format – see chapter 3.1
  - Unified-E (predefined): REST functions defined by Unified-E for extended read/write capabilities – see chapter 3.2
- Protocol: Web Protocol, HTTP or HTTPS
- Timeout (ms): The timeout value for calling the REST interface, in milliseconds.

Other specific parameters, depending on the selected web interface, are described below.

3 Web Server Interface for Datapoints

Basically, access to variables is symbolic. If no address is set for the datapoint in the Datapoints table in the Unified-E App Designer, the datapoint designation (column "Label") is used for addressing.

Depending on the "Web interface" adapter parameter, the endpoint adapter retrieves the datapoint values from the web server in different ways. The different REST interfaces are described in detail in the following subchapters.

3.1 Web Interface «Standard»

This JSON interface is already supported by many devices. The endpoint adapter can therefore read the desired datapoint values directly from the existing web server of the device (endpoint). Two types of access are supported to support the widest possible range of devices with a web server. The structure of the datapoint address depends on the type of access.

3.1.1 Access Type «Read JSON object»

With this type of access, individual values can be read from a large JSON object – a write is not provided.

A GET call returns an entire JSON object (e.g. many values of a device). The desired values of the JSON object can be mapped to datapoints.

The parameter 'Read URL path extension' is optional and allows a URL extension to the JSON object. To access the values it contains, the name or path to the JSON value must be entered in the Datapoints table under the "Address" column.

URL to the JSON object in the web server:

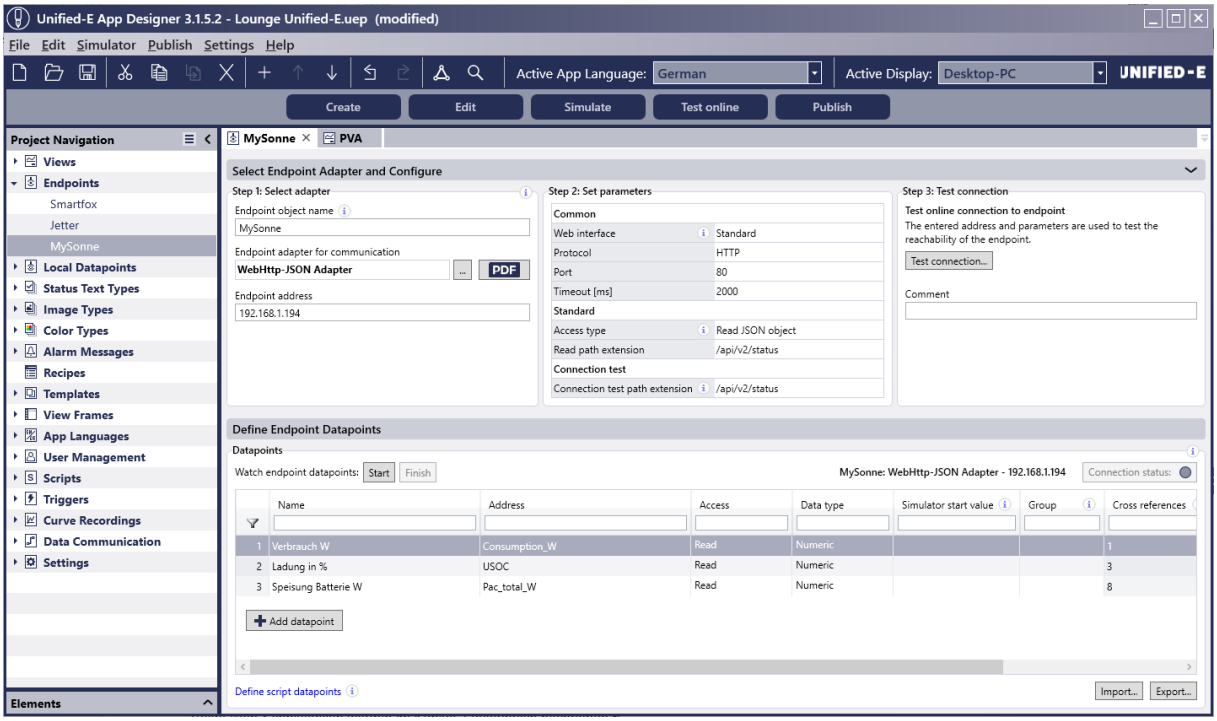
The URL pointing to the JSON object is built from the endpoint address and the parameters “Protocol” and “Read URL path extension” as follows:

<Protocol>://<Endpoint Address>:<Port>/<Read URL Path Extension>

Datapoint address (value in "Address" column):

Enter the path of the JSON value or the name, e.g. "Temperature" or "TemperatureSensor[0]. Value"

Example:



3.1.2 Access Type «Datapoint URL for «read/write»

For each datapoint, a GET request is used for reading, and a PUT or POST request is used for writing. The URL path extension to the datapoint is entered directly in the Datapoints table in the Address column.

URL to the datapoint:

The URL is composed as follows:

<Protocol>://<Endpoint Address>:<Port>/<Datapoint URL Path Extension (from Address column in table)>

URL example:

- Endpoint Address: 192.168.1.20
- Endpoint Parameters
    - Protocol: HTTP
    - Port: 80
    - Datapoint URL Path Extension: DataPoints
    - Address in datapoints table: "Sensor1/Temperature"

The datapoint is then read or written by the adapter at the following URL:

http://192.168.1.20:80/DataPoints/Sensor1/Temperture

Supported reading formats:

A simple JSON object of the form {<Name>: <value>} or a simple text/string with the value is expected.

Examples:

- {"Temperature": 30},
- 30

JSON format when writing:

{"<Datapoint URL Path Extension>": <value>}

Example: {"Temperature": 30}

Parameters «Read/Write request method»:

This parameter determines whether the PUT or POST action should be written.

3.2 Web Interface «Unified-E (predefined)»

With this web interface, the following REST actions are expected from the web server:

- GET action:
  - IsAlive: Endpoint reachability check
- PUT actions for datapoint access:
  - ReadValues: Read datapoints
  - WriteValues Write datapoints
  - ReadDataTables: Read table datapoints

These REST actions are specially optimized for communication with Unified-E and are described in Chapter 3.2.2 in more detail.

3.2.1 Specific Parameters

The following specific parameters are available for the endpoint adapter:

- Namspace: Also known as the resource path. Example URL: http://domain.com:80/Namensbereich/Aktion
- Authentication: Determines whether or authentication type should be used
  - No authentication: The web endpoint does not require authentication.
  - Custom authentication: Authentication occurs on each REST call. Username and password are carried with each call

- 'Http Basic Auth' authentication: The standard protocol 'Http Basic Auth' is used

- Datapoint Context: Determines whether the datapoint exists globally or per user:
  - Global: All users get the same value when they request a specific datapoint. This is typical for datapoints that represent machine states
  - Users: Different users can get different values when they request a specific datapoint. The endpoint needs to evaluate the username value of the JSON request
  - Language: All users of a language receive the same datapoint value for a given datapoint. This context is useful if datapoints of type “text* are to be multilingual. The REST action must evaluate the LanguageId for this

3.2.2 Required REST Actions

Common JSON properties:

The following JSON properties are available for all request actions:

- EndPointUser: User name for custom authentication
- EndPointPassword: Password for custom authentication
- User: The name of the app user or name of the registered operating device in the case of gateway communication. This value can be used by the web server to provide user-specific datapoints or values (see "Datapoint Context" parameters)
- LanguageId: Language code according to ISO 639-1 in 2-letter format (e.g. en for English, de for German). The web server can evaluate this value to return language-dependent content (see "Datapoint Context" parameter)

3.2.2.1 IsAlive Promotion

The GET call is called without any parameters and is used to check the basic reachability of an endpoint. The return value is expected to be true.

3.2.2.2 ReadValues Action

Reads multiple datapoints in a single REST call.

- HTTP Method: PUT
- Path: /ReadValues

Request (JSON):

```
{
  "EndPointUser": "admin",
  "EndPointPassword": "secret",
  "User": "john.doe",
```

```
  "LanguageId": "de",
  "DataPoints": [
    { "DataPointAddress": "Sensor1/Temperature" },
    { "DataPointAddress": "Sensor2/State" }
  ]
}
```

Response (JSON):

```
[
  {
    "Value": "22.5",
    "AccessState": 0
  },
  {
    "Value": "1",
    "AccessState": 0
  }
]
```

AccessState values:

- 0 – Value valid
- 1 – Datapoint not found
- 2 – Error

3.2.2.3 WriteValues Action

Writes new values for multiple datapoints.

- HTTP Method: PUT
- Path: /WriteValues

Request (JSON):

```
{
  "EndPointUser": "admin",
  "EndPointPassword": "secret",
  "User": "john.doe",
  "LanguageId": "en",
  "DataPointValues": [
```

```
    {
      "DataPoint": { "DataPointAddress": "Sensor1/Setpoint" },
      "Value": "45"
    },
    {
      "DataPoint": { "DataPointAddress": "Motor1/Start" },
      "Value": "1"
    }
  ]
}
```

Response (JSON):

"True" if successful, otherwise "False".

3.2.2.4 ReadDataTables Action

Reads tabular data from one or more datapoints (e.g., metrics).

- HTTP Method: PUT
- Path: /ReadDataTables

Request (JSON):

```
{
  "EndPointUser": "admin",
  "EndPointPassword": "secret",
  "User": "john.doe",
  "LanguageId": "en",
  "DataPoints": [
    {
      "DataPointAddress": "Logger1/Temperatures",
      "LastModificationId": "abc123"
    }
  ]
}
```

Response (JSON):

```
[
```

```
  {
    "DataTable": {
      "Rows": [
        { "Items": ["2024-01-01T10:00:00", "21.3"] },
        { "Items": ["2024-01-01T10:01:00", "21.4"] }
      ]
    },
    "ModificationType": 2,
    "AccessState": 0,
    "ModificationId": "def456"
  }
]
```

ModificationType values:

- 0 – All rows have been returned (reset)
- 1 – No changes since last query
- 2 – Added new lines at the top
- 3 – Added new lines at the bottom

AccessState values:

AccessState values see above.