



Pushing Performance



People | Power | Partnership

HARTING OPC-UA Gateway for MICA User Manual

1. Edition 2018

© HARTING IT Software Development, Espelkamp

All rights reserved, including those of the translation.

No part of this manual may be reproduced in any form (print, photocopy, microfilm or any other process), processed, duplicated or distributed by means of electronic systems without the written permission of HARTING IT Software Development GmbH & Co. KG, Espelkamp

Subject to alterations without notice.

Contents

1	The OPC-UA Gateway	2
1.1	About	2
1.2	OPC-UA	2
2	Installation	3
2.1	Installation of the OPC-UA Gateway	3
2.2	Basic container functionality	4
3	The OPC-UA Gateway user interface	5
3.1	The section "Devices"	6
3.2	The section "Executor"	8
3.3	The section "Reports"	10
4	The OPC UA Gateway REST interface	14
4.1	Quick test using a browser	14
4.2	Reading data using the Chrome® "Postman" add on	15
4.3	Writing data using the Chrome® "Postman" add on	16
4.4	Reading data using the wget shell command within the same MICA	16
4.5	Reading data using Node-RED within the same MICA	17

1 The OPC-UA Gateway

1.1 About

With the OPC-UA Gateway container it is possible to monitor multiple OPC-UA devices, as well as publishing that data to a set of subscriber endpoints and display them on a dashboard. The following user manual refers to the Firmware Base Version 3_523 and the Ha-VIS OPC-UA Gateway Container Version 2.0.0.

1.2 OPC-UA

The OPC Unified Architecture is a machine to machine communication protocol for industrial automation, which is a platform independent, service-oriented architecture that integrates all the functionality of the OPC Classic specifications into one extensible framework.

This multi-layered approach accomplishes the original design specification goals of:

1. **Functional equivalence:** all COM OPC Classic specifications are mapped to UA
2. **Platform independence:** from an embedded micro-controller to cloud-based infrastructure
3. **Secure:** encryption, authentication, and auditing
4. **Extensible:** ability to add new features without affecting existing applications
5. **Comprehensive information modeling:** for defining complex information



For further information about the OPC Unified Architecture refer to:
<https://opcfoundation.org/>

2 Installation

You may skip this chapter, if the "OPC-UA Gateway" container is already installed.

2.1 Installation of the OPC-UA Gateway

Log in to the MICA as admin and click the "Install" icon.

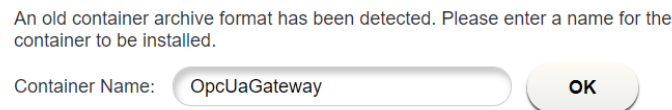


Install

On the next view you can choose the installation archive of the OPC-UA Gateway . Next click the Execute button.



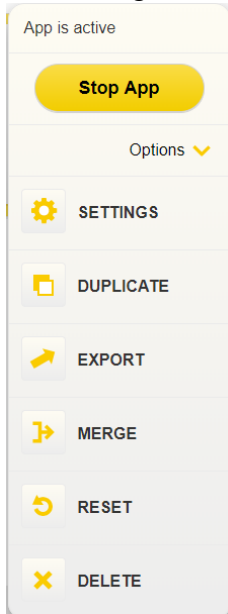
A dialog will open up where you can name the container you want to upload. When you are finished naming, click OK to install the container.



As soon as the icon is visible you can start the container.

2.2 Basic container functionality

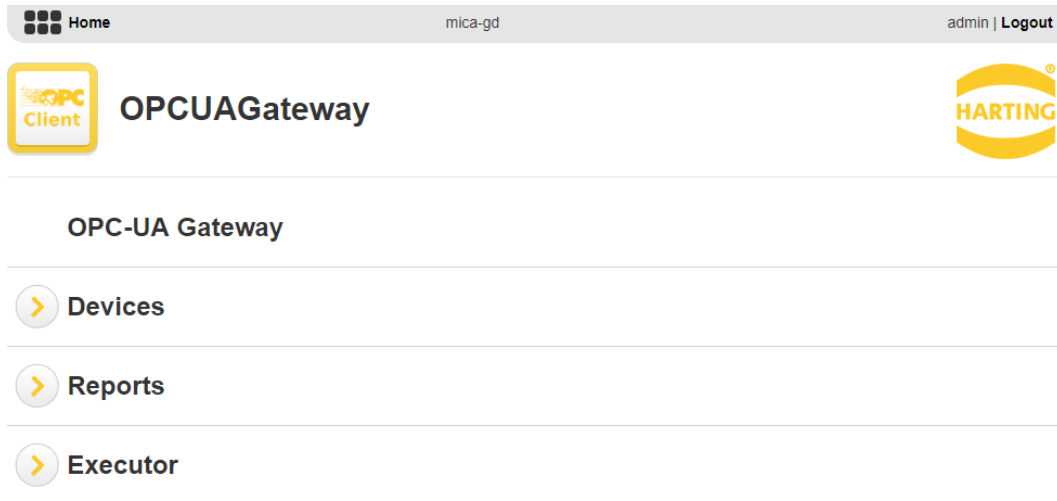
By executing an right click in the OPC-UA Gateway container icon a menu will appear containing the following functions.



1. **Settings** lets you update the OPC-UA Gateway container.
2. **Duplicate** lets you duplicate the container on your MICA.
3. **Export** lets you export the container to your PC. All configurations you set will be kept.
4. **Merge** merges the overlay and underlay of the container.
5. **Reset** resets the configuration of the OPC-UA Gateway container to factory default.
6. **Delete** lets you delete the container.

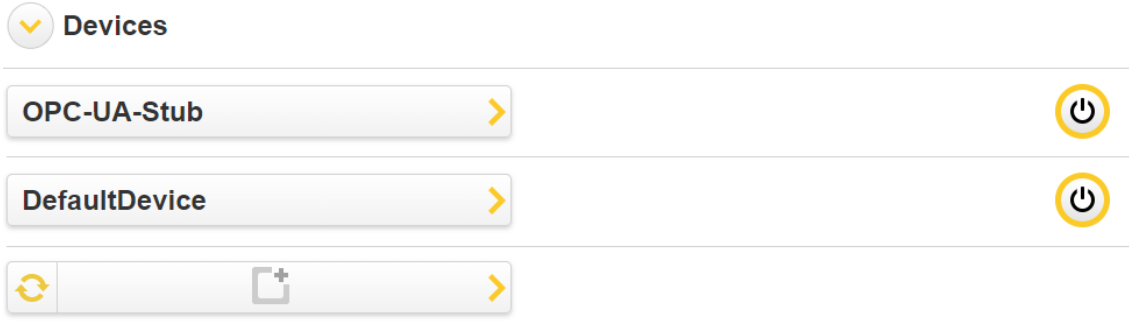
3 The OPC-UA Gateway user interface


With a click on the *OPC-UA Gateway* container icon the user interface of the gateway appears. Here 3 different sections are available which provide different functionalities.




3.1 The section “Devices”

In the section *Devices* you can create, edit and import OPC-UA devices from which you want to collect data.







By clicking the add button  a window will appear containing the following fields:


Device


Name


Period **ms**

 **Properties**

device.host	▼	127.0.0.1	
device.username	▼	username	
device.password	▼	
Name		Value	

 **Fields**

Accept

1. **Name** is the displayed name of the device. The name will be also displayed in the section *Reports and Executor*.
2. **Period** is the sampling rate in milliseconds.
3. **Properties** is a list for defining mandatory and optional properties. Mandatory for the connection is the host. There are several optional properties like the port or credentials for authorization. Properties can be added with a click on the *Name — Value* button or deleted with a click on .

By clicking the “Fields” button a field list will appear. When you click in the box a new dialog appears where you can create a field for the device with the following properties.

Field

Name

Properties

field.identifier	▼	identifier	⊗
field.identifier.type	▼	string	▼ ⊗
field.period	▼	5000	⊗
Name		Value	

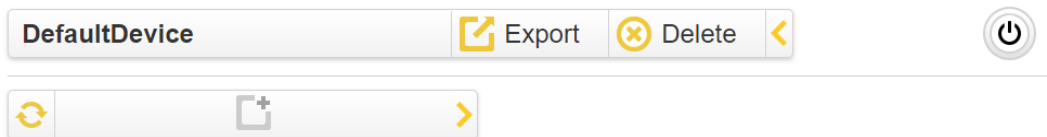
Accept


1. **Name** is the name of the field. It will also be displayed in the section *Reports*.
2. **Properties** is a list for defining mandatory and optional properties for the field. The mandatory properties are *identifier*, which sets the identifier for a node in the address space of an OPC-UA server and *type*, which sets the type of the nodeId. The supported types are string and numeric.
Optionally the *period* property can be set to define the sampling rate for the field in milliseconds.

If you have created the fields you need and entered all mandatory properties click on the “Accept” button to add your device to the list.

3.1.1 Import/Export Devices

If you hover over a device, a menu will appear on the right side. The menu contains the two options *Export* and *Delete*. Deleting the device will remove it from the list. You can export the device to save the current configuration or to create a template for same devices so you only have to change the name.



To import a device hover over the  button. A menu with an import option will appear. Click on that button and choose the device you want to import and click open. The device will be shown in the list.

3.2 The section "Executor"

The "Executor" is a small tool to easily read data from or write data to devices which are connected to the OPC-UA Gateway.

▼ **Executor**

Device URL	<input type="text" value="opc.tcp://0.0.0.0:4840"/>	▼
Identifier	<input type="text"/>	
Operation	<input type="text" value="READ"/>	▼
Identifier Type	<input type="text" value="string"/>	▼
Result	<input type="text"/>	
Data	<input type="text"/>	

Execute

To execute a OPC-UA query you have to provide 4 properties:

Device URL defines the device you want to connect to. You can either choose an active device from the section *Devices* or enter the address of an OPC-UA device. If your connection requires an other port than the default port 4840 you have to add the port to the connection string as shown in the example.

Identifier indicates a node in the address space of an OPC-UA device. If you have selected a device as the "Device URL" you are also able to choose from the defined fields of that device.

Operation Select either "READ" or "WRITE".

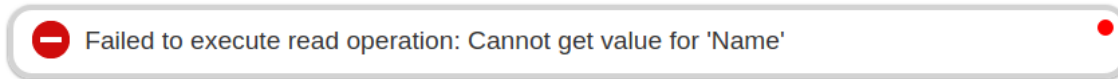
Identifier Type is the format and data type of the identifier. It can be a numeric value or a string. This property will only be visible if no field was selected as the "Identifier".

Run the Executor by clicking the **Execute** button.

3.2.1 Error messages

When executing an OPC-UA query you may run into different error messages. The most important ones will be elucidated in this chapter.

An OPC-UA error message will appear on the bottom of the screen.



An OPC-UA error message consists of the parts. The first part is telling you what operation has failed, i.e. *Failed to execute read operation:*. The second part tells you more information on what caused the problem. The following list shows the main mistakes that can lead to an error:

1. **Cannot get value for 'Identifier'**

This message will appear if the OPC-UA Gateway was not able to establish a connection to the given "Device URL".

2. **Cannot get value for 'Identifier': INVALID_PARAMETER**

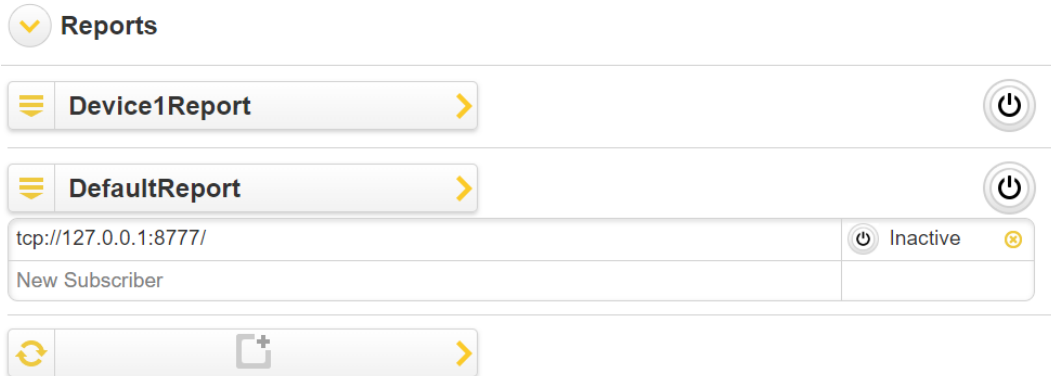
This message will appear if the OPC-UA Gateway was not able to get a value for the given "Identifier".

3. **Scheme seems invalid! Supported scheme is 'tcp'**

This message will appear if you have used an unsupported scheme in the "Device URL".

3.3 The section "Reports"

The section "Reports" is for publishing machine data on a "Subscriber" endpoint.



If you want to add a new report click on . A dialog will appear where you can enter the following information:

Report

Name

Duration

Repeat Period

Fields

DefaultDevice	▼	DeviceName	▼	
DefaultDevice	▼	DeviceManual	▼	
DefaultDevice	▼	DeviceStatus	▼	
DefaultDevice	▼	DeviceInfo	▼	
DefaultDevice	▼	Temperature	▼	
DefaultDevice	▼	Temperature2	▼	
Device		Field		

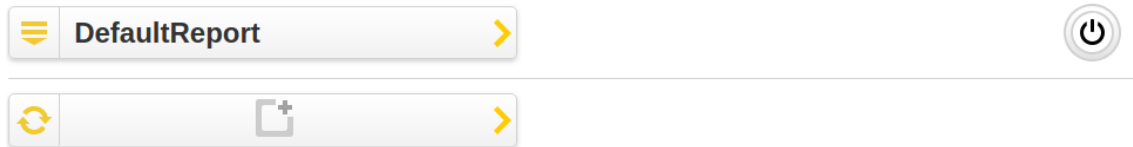
Accept


1. **Name** is the displayed name of the report.
2. **Duration** is the duration of the reports in ms, i.e. how long the report should run.
3. **Repeat Period** is the time interval for the next report to start. As long as the repeat period is smaller as the duration the start of the next report is immediately after the last.
4. **Fields** is for choosing your device you added in the section *Devices* and the field you want to observe. You can add new fields by clicking on the *Device — Field* button.

If you have entered all needed information click on Accept to add the report.


3.3.1 Import/Export Reports

If you hover over a report, a menu will appear on the right side. The menu contains the two options *Export* and *Delete*. Deleting the report will remove it from the list. You can export the report to save the current configuration or to create a template for some reports so you only have to change the name.




To import a report hover over the  button. A menu with an import option will appear. Click on that button and choose the report you want to import and click open. The report will be shown in the list.

3.3.2 Subscriber

When you have added the report you can click on  and a drop-down menu will appear where you can add a new subscriber endpoint. The supported types of endpoints are HTTP, HTTPS, MQTT, MQTTS, TCP, UDP and AZURE.

For adding an new subscriber click in the field textitNew Subscriber and a dialog will appear where you can choose the type of the endpoint and enter all needed information. You find an example for a subscriber endpoint in the picture below.

Subscriber


Type

MQTT
▼

Host

MQTT-mica-eg2dy.local

Port

1883

Topic

/opcua

Client ID

gateway

QoS

Exactly once
▼

User Info

URI

mqtt://MQTT-mica-eg2dy.local:1883/opcua?clientid=

Properties

Name	Value

Accept

3.3.3 Report Example

If you have configured your “Devices”, “Reports” and your subscriber endpoint, you will get a report after the time which was specified by you.

A Report is a JSON structure containing the following fields:

Report
<pre> termination : String; terminator : String; devices : Device[]; applicationId : String; initiator : String; initiation : String; reportName : String; date : String; totalMilliseconds : String; </pre>

The fields of the Reports are specified in the following table:

Field	Type	Description
termination	String	The termination condition of the report.
terminator	String	The terminator of the report.
devices	Device[]	All monitored devices.
applicationId	String	The ID of the application.
initiator	String	The initiator of the report.
initiation	String	The initiation condition of the report.
reportName	String	The name of the report.
date	String	The creation datetime of the report.
totalMilliseconds	String	The duration in which data was collected.

You find an example of a report in the field below:

```
{
  "applicationId":"OpcUaGateway",
  "reportName":"DeviceReport",
  "date":"2018-18-05T08:35:04.843",
  "totalMilliseconds":1000,
  "initiation":"REQUESTED",
  "initiator":null,
  "termination":"DURATION",
  "terminator":null,
  "devices":[
    {
      "name":"Device 1",
      "fields":[
        {
          "name":"Temperature",
          "value":43.5906,
          "date":"2018-05-16T21:13:03.959"
        },
        {
          "name":"Maximum injection pressure",
          "value":534.5047,
          "date":"2018-05-16T21:13:03.974"
        },
        {
          "name":"Injection time",
          "value":0.41,
          "date":"2018-05-16T21:13:03.990"
        },
        {
          "name":"Heating zone 1",
          "value":265.0,
          "date":"2018-05-16T21:13:04.005"
        }
      ]
    }
  ]
}
```

4 The OPC UA Gateway REST interface

The " OPC UA Gateway App" provides a REST interface" which allows gathering information from the gateway and its connected devices without the need of using the OPC UA protocol.

For an automatic access to the gateway it is essential to provide authorization data normally included in the request headers of the commands. The needed encrypted content can be copied e.g. from the Chrome[®] add on "Postman" as shown in the following code snippets. In case of a newly started browser, you will be requested to enter your credentials prior to the first execution of the REST command.



Make sure using the IP address of the MICA base when calling the REST interface from a client outside the MICA.

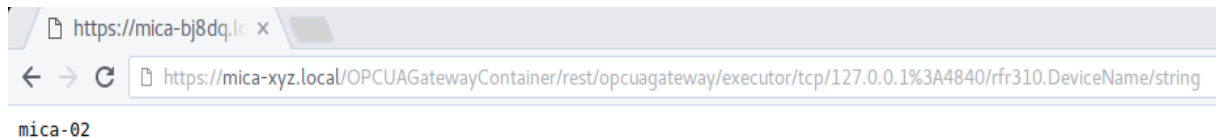
If you call the interface from a container within the same device you can use the non secured http-protocol to save access time.

4.1 Quick test using a browser

By means of a simple GET command you can retrieve information from the device using a simple GET command. Enter the following command in a browsers command-line:

```
https://mica-xyz.local/[CONTAINER_NAME]/rest/opcuagateway/executor/tcp/127.0.0.1%3A4840/rfr310.DeviceName/string
```

You will see:



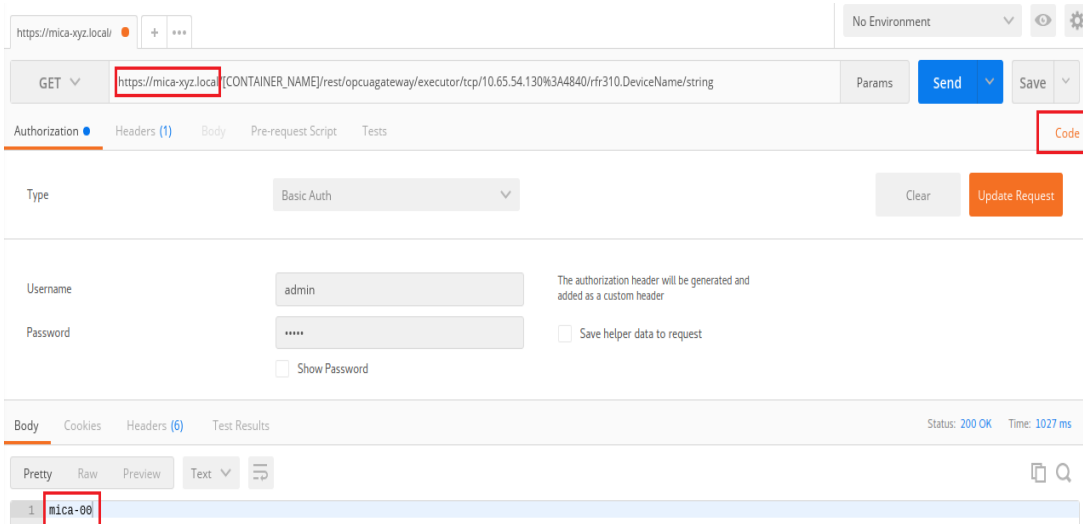
In the command-line you will find all the details you also have to specify in the Executor". The device address of the MICA base `https://mica-xyz.local/` is followed by: - The path to the executor section of the gateway `/[CONTAINER_NAME]/rest/opcuagateway/executor/`, followed by:

- The connection type **tcp/**, followed by:
- The [IP address]:[PORT] **127.0.0.1%3A4840/**, followed by:
- The field identifier **rfr310.DeviceName/**, followed by:
- The the identifier type **string**, followed by:

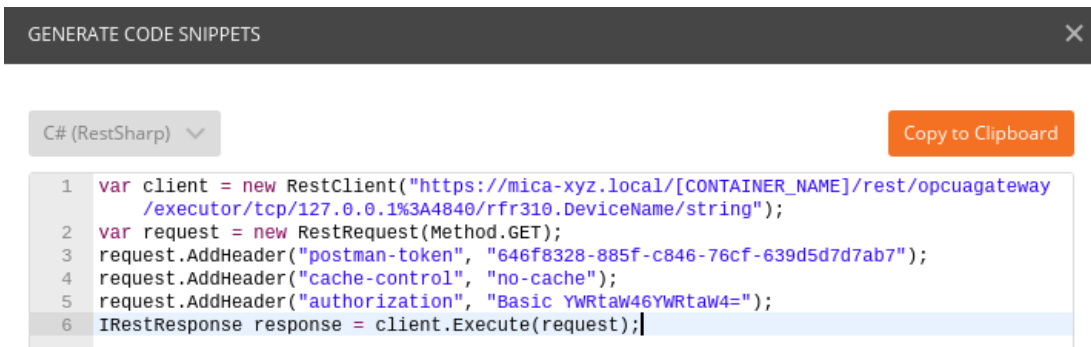
The Operation mode (READ) is automatically selected by means of the GET command.

4.2 Reading data using the Chrome® "Postman" add on

Software like the Chrome® add on "Postman" can also be used to send GET commands. It gives a better overview on how the commands behave and provides more details.



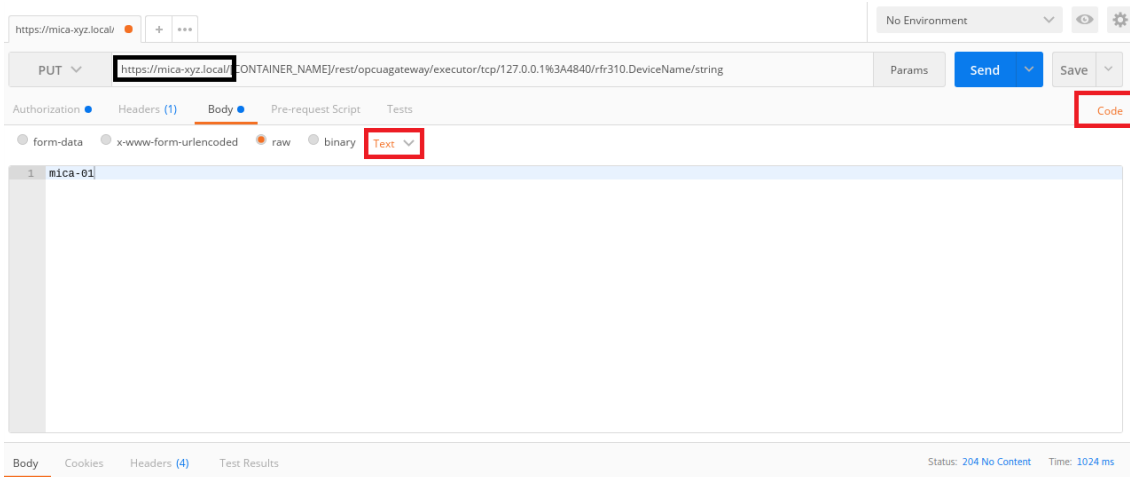
On clicking to "Code", Postman generates code snippets for using the REST commands in many other programming languages, here C#:



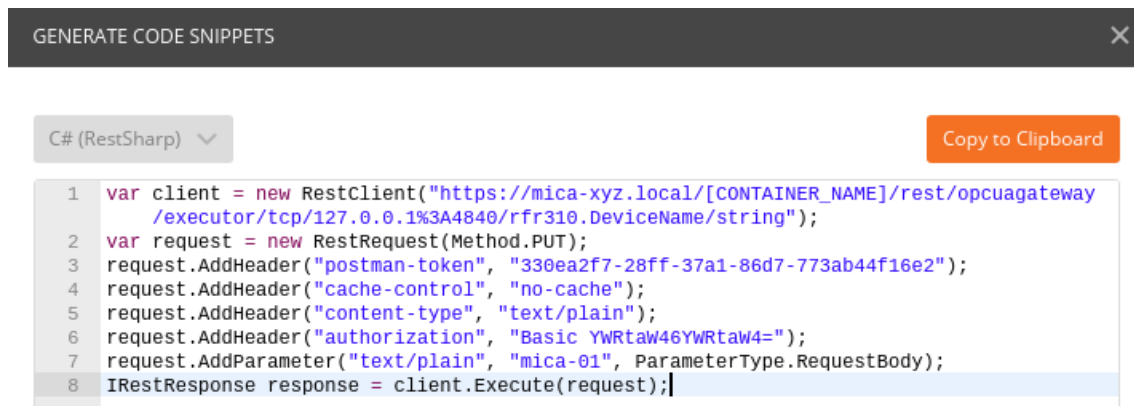
4.3 Writing data using the Chrome® "Postman" add on

By means of a PUT command you may write data to the field identifier. To test it, use again the Chrome® add on "Postman" and enter e.g.

`https://mica-xyz.local/[CONTAINER_NAME]/rest/opcuagateway/executor/tcp/127.0.0.1%3A4840/rfr310.DeviceName/string`



Here too, a generated code snippet for C#:



4.4 Reading data using the wget shell command within the same MICA

As mentioned at the beginning of that chapter, you may also use the REST interface between two different containers of the same device. Using the shell command "wget" is a very useful approach for a quick test of the REST interface.

The sample script below accepts on parameter for the IP address, stores the result in a variable and prints it on the console.

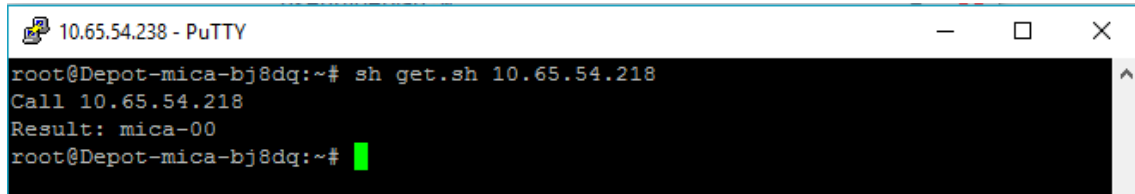
```

#!/bin/sh
IP=$1
RESTPATH="rest/opcuagateway/executor/tcp/10.65.54.130/rfr310.DeviceName/string"
printf "Call: %s\n" $IP
RESULT="`wget -qO- http://admin:admin@$IP/$RESTPATH`"
printf "Result: %s\n" $RESULT

```

Using an ssh software like "Putty" log yourself in to another container as the Apps container and create a new file called "get.sh". Make it executable and call it using the syntax: sh get.sh 10.65.54.218.

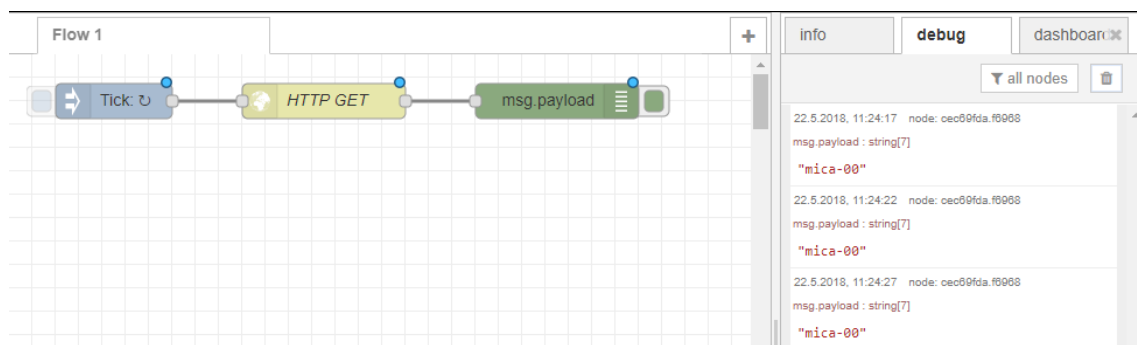
The output should look like:




```
10.65.54.238 - PuTTY
root@Depot-mica-bj8dq:~# sh get.sh 10.65.54.218
Call 10.65.54.218
Result: mica-00
root@Depot-mica-bj8dq:~#
```

4.5 Reading data using Node-RED within the same MICA

Also the Node-RED container available at <https://www.harting-mica.com/> can be easily used to establish connections to the Gateway Containers REST interface. Start Node-RED and use the inject, the http request and the debug node to compose a scenario sending permanent Http-GET commands to the Gateway.



Here you may download the sample flow:  [httpget.json](#)