GlueLogics IoT Software
REST API and Websocket Short Reference Manual

Confidential:

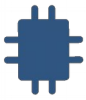
All information in this document is subject to confidentiality and only for the eyes of involved persons.

eptecon

Germany

www.eptecon.com
info@eptecon.com

Author: EP
Version: 0.9
Date: 11/22/20



Version history

Version	Change	Author	Reviewer	Publisher	Valid from
0.9	Draft	EPR			11/22/2020

Table of Contents

1 Introduction

1.1 Scope

The scope of this document is to provide a short description of the most important REST API or Websocket commands of the GlueLogics IoT Software. The document addresses the developers of applications that require communication with the GlueLogics IoT Software to make use of device data. The reference manual uses examples from the currently running Industrial IoT Scales Project.

1.2 GlueLogics IoT Software

GlueLogics IoT Software is a Java Application for data collection from IoT devices with different sensors. Besides, it allows data preprocessing, visualization and IoT devices provisioning and management. It uses Cassandra noSQL data base for data persistency (PostgreSQL is also possible) and provides data to other server-side applications via REST API or Websocket. The devices can be provisioned and managed using GlueLogics IoT Software Web Interface by users. On the device side GlueLogics IoT Software utilizes MQTT(s) or HTTP(s) for data collection. GlueLogics IoT Software is developed to run on Linux based IoT Gateways and Cloud servers with Java run time engine. Figure 1 shows the architecture of the GlueLogics IoT Software.

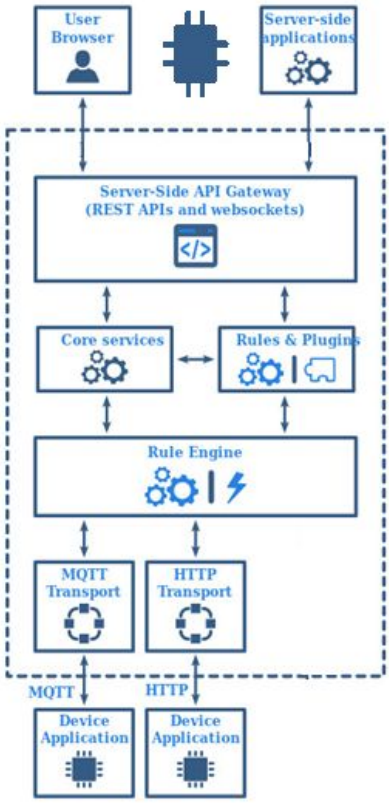


Figure 1: GlueLogic Architecture

2 Prerequisites and Structures

2.1 IoT Gateway Hardware

For the use of GlueLogics IoT Software on IoT Gateway it shall be installed on a Single Board Computer (e. g. Raspberry Pi 3) with at least 1 GB RAM and a Debian based Linux distribution. Further, Java 8 (e. g. OpenJDK) is required to run GlueLogics IoT Software.

2.2 Entity structures

GlueLogics IoT Software works with entities, which can be defined as follows:

- Tenants: Software administrators,
- Customers: IoT devices and software users,
- Assets: Physical places, where IoT devices are used,
- Devices: IoT devices with different sensor types.

2.3 Data Structures

All IoT devices connected to GlueLogics IoT Software have following basic data structure:

- Attributes: device specific data which are constantly saved on the device, comprising
  - “keys”, which represent the data description and
  - “values” which represent the value of the according data key,
- Telemetry data: device specific data gathered by the specific sensor, comprising
  - “keys”, which represent the data description and
  - “values” which represent the value of the according data key.

The device data consist of multiple “key”- “value” pairs and are provided as JSON objects from device. Each “key”- “value” pair is enriched with UNIX timestamp when received by the GlueLogics IoT Software. Following example shows attributes and telemetry data JSON objects as used in the Industrial IoT Scales.

Attributes:

```
[
	{
		"lastUpdateTs":1605574422006,
		"key":"FWversion",
		"value":"1.01.00a"
	},
	{
		"lastUpdateTs":1605574422006,
		"key":"clientID",
		"value":"test_client"
```

```
        },
        {
                "lastUpdateTs":1605574422006,
                "key":"deviceID",
                "value":"Scale001"
        },
        {
                "lastUpdateTs":1605574422006,
                "key":"serialNumber",
                "value":"1001-00001"
        },
        {
                "lastUpdateTs":1605574422006,
                "key":"typeClass",
                "value":"SmartScale"
        }
]
```


Telemetry:


```
{
        "weightValue":
        [
                {
                        "ts":1605574822333,
                        "value":"1.023"
                }
        ],
        "weightUnits":
        [
                {
                        "ts":1605574822333,
                        "value":"kg"
                }
        ],
        "localTemperature":
        [
                {
                        "ts":1605574822333,
                        "value":"22.250"
                }
        ],
        "batteryVoltage":
        [
                {
                        "ts":1605574822333,
                        "value":"118"
                }
        ]
}
```

2.4 Required credentials

Additionally, following credentials are required to access the GlueLogics IoT Software via REST API or Websocket:

- Customer ID (defined by GlueLogics IoT Software when setting up IoT Gateway, e. g. 19d0dc10-2db9-11eb-b73b-7350bf54db8a),
- Authentification Token (JSON Web Token),
- Device ID (defined by GlueLogics IoT Software when provisioning customer specific devices, e. g. 40b799e0-286e-11eb-8145-7350bf54db8a).

3 REST API Reference

3.1 Token based authentication

Get authentication token for REST API connection:

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{"username":"admin@qube-ing.de", "password":"qubeadmin"}' 'http://$GLUELOGICS_URL:8080/api/auth/login'
```

Replace $GLUELOGICS_URL with the URL of the server, where GlueLogics IoT Software is running. If your application runs on the same server, replace it with “localhost”.

On success you will get following response data:

```
{"token":"$JWT_TOKEN", "refreshToken":"$YOUR_JWT_REFRESH_TOKEN"}
```

3.2 Customer devices overview

Get customer devices overview:

```
curl -v -X GET http://localhost:8080/api/customer/$CUSTOMER_ID/devices?limit=$DEVICE_COUNT \
--header "Content-Type:application/json" \
--header "X-Authorization: Bearer $JWT_TOKEN"
```

Replace $CUSTOMER_ID and $JWT_TOKEN with the Customer ID as described in 2.4 and token received after authentication request respectively. Replace $DEVICE_COUNT with the number of provisioned devices.

On success you will get following response data (example for one device):

```
{
	"data":
	[
		{
			"id":
```

```
                        {
                                "entityType":"DEVICE",
                                "id":"=$DEVICE_ID"
                        },
                        "createdTime":1605573956222,
                        "additionalInfo":null,
                        "tenantId":
                        {
                                "entityType":"TENANT",
                                "id":"=$TENANT_ID "
                        },
                        "customerId":
                        {
                                "entityType":"CUSTOMER",
                                "id":"=$CUSTOMER_ID "
                        },
                        "name":"device name",
                        "type":"Device Type"
                }
        ],
        "nextPageLink":null,
        "hasNext":false
}
```


3.3 Data from devices

Get device attributes for a connected Device:

```
curl -v -X GET http://localhost:8080/api/plugins/telemetry/DEVICE/$DEVICE_ID/values/attributes?keys=key1,key2,key3 \
--header "Content-Type:application/json" \
--header "X-Authorization: Bearer $JWT_TOKEN"
```

Replace $DEVICE_ID with the Device ID for the specific device and $JWT_TOKEN with the token received after authentication request. Replace key1, key2 etc. with the desired device attributes keys (e. g. “deviceID”).

On success you will get response data as shown in the example in 2.3.

Get Device Telemetry for a connected Device:

```
curl -v -X GET http://localhost:8080/api/plugins/telemetry/DEVICE/$DEVICE_ID/values/timeseries?keys=key1,key2,key3 \
--header "Content-Type:application/json" \
--header "X-Authorization: Bearer $JWT_TOKEN"
```

Replace $DEVICE_ID with the Device ID for the specific device and $JWT_TOKEN with the token received after authentication request. Replace key1, key2 etc. with the desired device telemetry keys (e. g. “WeightValue” for the Industrial IoT Scales).

On success you will get response data as shown in the example in 2.3.

4 Websocket Reference

4.1 Token based authentication

For the websocket authentication the same authentication method is used as described in 3.1.

4.2 Opening websocket connection

Websocket connection to subscribe for telemetry data:

ws://localhost:8080/api/ws/plugins/telemetry?token=$JWT_TOKEN

Replace $JWT_TOKEN with the token received after authentication request.

On success a websocket connection will be opened.

4.3 Subscribing to telemetry data

Subscription for the telemetry data with defined data keys:

```
{"tsSubCmds":
[{"entityType":"DEVICE","entityId":"$DEVICE_ID","keys":"key1,key2,key3","cmdId":1}],"
historyCmds":[],"attrSubCmds":[]}
```

Replace $DEVICE_ID with with the Device ID for the specific device. Set the unique command ID (cmdId) for the opened websocket connection. Replace key1, key2 etc. with the desired device telemetry keys (e. g. “WeightValue” for the Industrial IoT Scales).

On success device telemetry data will be sent. It will be resent every time GlueLogics IoT Software receives the device telemetry data from the specific device.