Arrowhead Historian Service IDD

**Abstract**

This document defines interface design of the Historian Service based on the REST approach within Arrowhead Framework generation 4.1.3.

1. Overview

This document describes the Historian Service of G4.1.3 with a REST-based interface.

This interface can use HTTP, HTTPS, CoAP, and CoAPS.

1. Interfaces

As per the SD of this Service, there is four methods implemented. Table 1 describes these methods. ListSystems returns a list of all unique systems that have data stored. ListServices lists all services per system that have stored data. DataStore stores new data while DataFetch returns already stored data.

Table 1 Function description

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **URL subpath** | **Method** | **Input/Output** | **Output** |
| ListSystems | /historian | GET | **SystemList** | OK, NOT\_AUTH INTERNAL\_SERVER\_ERR |
| ListServices | /historian/{systemName } | GET | **ServiceList** | OK, NOT\_AUTH INTERNAL\_SERVER\_ERR |
| DataStore | /historian/{systemName }/{serviceName} | PUT | **SensorData** | OK, BAD\_REQUEST NOT\_AUTH INTERNAL\_SERVER\_ERR |
| DataFetch | /historian/{systemName }/{serviceName} | GET | **SensorData** | OK, BAD\_REQUEST  UNAUTHORIZED NOT\_FOUND  INTERNAL\_SERVER\_ERR |

1. Information Model

The SensorData request payload contains generic sensor data. SensorData normally contains information about the unit, source system, timestamp and metadata. The default payload type is JSON-encoded SenML (RFC 8428). The response to a Store/ Retrieve request is a simple HTTP/CoAP status code (Created/OK – request was success, No Content – request had no effect). For the Push function, the content-type must be set to ‘application/json’. For Fetch, the response content-type is ‘application/json’.

# 3.1. ListSystems

# 3.2. ListServices

# 3.3. SensorData Store- SenML request

The message below is an example of a SenML formatted message that that is valid input to the Historian service. A consuming system can store SenML formatted data.

[

{"bn":" TempSys23433:","bt":1.276020076001e+09, "bu":"Cel","bver":5,

"n":" IndoorTemperature ","u":"Cel","v":22.1},

{"n":" IndoorTemperature ","t":-5,"v":22.2},

{"n":" IndoorTemperature ","t":-4,"v":22.4}

]

# 3.4. SensorData Fetch- SenML response

The message below is an example of a SenML formatted message that that is valid out from the Historian service. A consuming system can fetch SenML formatted data.

[

{"bn":" TempSys23433:","bt":1.276020076001e+09, "bu":"Cel","bver":5,

"n":" IndoorTemperature ","u":"Cel","v":22.1},

{"n":" IndoorTemperature ","t":-5,"v":22.2},

{"n":" IndoorTemperature ","t":-4,"v":22.4}

]

When a consumer performs a standard GET request, the Historian will reply with SenML encoded data. There are parameters that can be used to control

1. Revision history

# Amendments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Date | Version | Subject of Amendments | Author |
| 1 | 2018-09-17 | G4.0 | Initial | Jens Eliasson |
| 2 | 2018-10-30 | G4.0 | Text update | Jens Eliasson |
| 3 | 2019-03-20 | G4.0 | Updated data model to RFC 8428 | Jens Eliasson |
| 4 | 2020-03-27 | G4.1.3 | Updated models and text | Jens Eliasson |
| 5 | 2020-11-13 | G4.1.3 | Added more information | Jens Eliasson |
|  |  |  |  |  |

# Quality Assurance

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Date | Version | Approved by |
| 1 |  |  |  |
| 2 |  |  |  |