Arrowhead DataManager Core Service 4.1.3 Release Notes

In Arrowhead version 4.1.3, the whole codebase of the Arrowhead framework has been rewritten using Spring Boot technology. We believe the new version is better both in code quality, performance and robustness. It is easier to use, maintain and improve.
However, this means that Arrowhead version 4.1.3 is NOT compatible with providers and consumers written for version 4.1.2 because the changes affect not just the backend but the public interface of the core services, too.
This document describes the new public API of the Arrowhead DataManager Core Service 4.1.3 and shows the key differences between the new interface and the old one.

Client Services

These services can be used by consumers in an Arrowhead Cloud.

Uri: /datamanager/echo
Type: GET

Returns a “Got it” message with the purpose of testing the core service availability.

Uri: /datamanager/proxy
Type: GET

Gets a list of all systems that have created a proxy endpoint..

Output JSON structure for empty set:

```
{
  "systems":[]
}
```

or

```
{
  "systems":["sys1"]
}
```

if one system with the name ‘sys1’ has created a service endpoint.

Uri: /datamanager/proxy/
Type: PUT

Depending on input message, this operation can list, create or delete an endpoint for a specific service. The endpoint can then be used by a service data producer to push SenML data to the Proxy service.

Input JSON structure for List operation:

```

{
  "op":"list"
}
```

Input JSON structure for Create operation:

```
{
  "op":"create",
    "srvName": "_temperature._http._sys1._arrowhead.eu",
    "srvType": "_temperature"
}
```

Input JSON structure for Delete operation:

```
{
  "op":"delete",
  "srvName": "_temperature._http._sys1._arrowhead.eu"
}
```

Uri: /datamanager/proxy/<systemName>
Type: GET

Gets a list of all all services that system <systemName> have created.

Output JSON structure for empty set:

```
{
  "services":[]
}
```

or

```
{
  "services":[
    "_temperature._http._sys1._arrowhead.eu"
  ]
}
```

when one system named *sys1*, has registered one service. The example request URI is: /datamanager/proxy/sys1

Uri: /datamanager/proxy/<systemName>/<serviceName>
Type: PUT
Content-Type: application/json

Pushes a SenML message to a service endpoint named <serviceName> that system <systemName> have created. Building on the previously used examples with sys1 as system, the entire URI will be: /datamanager/proxy/sys1/_temperature._http._sys1._arrowhead.eu

Input JSON:

```
[
  {"bn":"_temperature._http._sys1._arrowhead.eu", "bt": 14625223},
  {"n": "temperature", "v": 19.2}
]
```

4.1.3.

- Note that the *bn* must occur once, and must be placed in the first JSON object in the array.
- Note that the *bt* must occur once, and must be placed in the first JSON object in the array. The bt tag is the number of seconds since 1/1 1970, i.e. UNIX time.
- Note that the *bu* can only occur once, and if exists, must be placed in the first JSON object in the array.
- The *bn* field must match the serviceName parameter.
- For more details, see the SenML RFC - https://tools.ietf.org/html/rfc8428

Uri: /datamanager/proxy/<systemName>/<serviceName>
Type: GET
Content-Type: application/json

Fetches the lastest SenML message from a service endpoint named <serviceName> belonging to system <systemName>.

Output JSON:

```
[
  {"bn":"_temperature._http._sys1._arrowhead.eu", "bt": 14625223},
  {"n": "temperature", "v": 19.2}
]
```

Note that the Proxy service only stores one message (the newest one), so if a data producer uploads messages faster than a consumer is downloading them, data will be used. Use the Historian service if data guarantees are needed.

4.1.3.

- *authorization* – Subscribers will only receive events form publishers if the subscriber system has a valid authorization record with the publisher (subscriber as consumer, publisher as provider, for any service and interface).
- *sources* – If sources field is empty or not present the subscriber will receive events from all authorized publishers. If there is even one publisher system in the sources field, the subscriber will only receive events from the authorized publishers from the given publisher systems. Unlike the 4.1.2 version, if a given publisherSystem do not exists in the database it will not be created .
- *matchMetaData* – If matchMetaData field is true, filterMetaData field must have at least one key-value pair defined.
- *filterMetaData* – If filterMetaData field has key-value pairs defined and matchMetaData field is true, the subscriber will only receive events when the event has the all the key-value pairs defined in its metadata. The keys must match case, the values are not case sensitive.
- *startDate* – If startDate is defined, the subscriber system will only receive events when the events timestamp is after startDate. StartDate must be after the current datetime.
- *endDate* – If endDate is defined, the subscriber system will only receive events when

the events timestamp is before endDate. EndDate must be after the current datetime. If startDate is defined endDate must be after startDate.

- subscriberSystem – Unlike the 4.1.2 version, if a given subscriberSystem do not exists in the database it will not be created .

Uri: /eventhandler/unsubscribe
Type: DELETE
Query params:

- *event_type* – string value of subscriptions event type name (mandatory),
- *system_name* – string value of subscriber system name (mandatory),
- *address* – string value of subscriber system address (mandatory),
- *port* – integer value of subscriber system port (mandatory)

Removes the subscription record specified by parameters.

The following services are no longer exist:

- DELETE /eventhandler/subscription/type/{type} /consumer/{systemName}
- PUT /eventhandler/subscription

Management Services

These services can only be used by the system operator of the local cloud. All date fields contain the text representation of a UTC timestamp.

Uri: /eventhandler/mgmt/subscriptions
Type: GET
Query params:

- *page* – zero-based page index (optional),
- *item_per_page* – maximum number of items returned (optional),
- *sort_field* – sort field (optional, default: id, possible values: id, createdAt, updatedAt),
- *direction* – direction of sorting (optional, default: ASC, possible values: ASC or DESC)

Returns a page of subscription record. If page and item_per_page are not defined, returns all records.

Returned JSON structure:

```
{
  "count": 0,
  "data": [
    {
      "id": 0,
      "eventType": {
        "id": 0,
        "eventTypeName": "string",
        "createdAt": "string",
        "updatedAt": "string"
      },
      "filterMetaData": {
        "additionalProp1": "string",
        "additionalProp2": "string",
        "additionalProp3": "string"
      },
      "matchMetaData": true,
      "notifyUri": "string",
      "sources": [
        {
          "id": 0,
          "systemName": "string",
          "address": "string",
          "authenticationInfo": "string",
          "port": 0,
          "createdAt": "string",
          "updatedAt": "string"
        }
      ],
      "startDate": "string",
      "endDate": "string",
      "subscriberSystem": {
        "id": 0,
        "systemName": "string",
        "address": "string",
        "authenticationInfo": "string",
        "port": 0,
        "createdAt": "string",
        "updatedAt": "string"
      },
      "createdAt": "string",
      "updatedAt": "string"
    }
  ]
}
```

Uri: /eventhandler/mgmt/subscriptions/{id}
Type: GET

Returns the subscription record specified by the id path parameter.

Returned JSON structure:

```
{
      "id": 0,
      "eventType": {
        "id": 0,
        "eventTypeName": "string",
        "createdAt": "string",
        "updatedAt": "string"
      },
      "filterMetaData": {
        "additionalProp1": "string",
        "additionalProp2": "string",
        "additionalProp3": "string"
      },
      "matchMetaData": true,
      "notifyUri": "string",
      "sources": [
        {
          "id": 0,
          "systemName": "string",
          "address": "string",
          "authenticationInfo": "string",
          "port": 0,
          "createdAt": "string",
          "updatedAt": "string"
        }
      ],
      "startDate": "string",
      "endDate": "string",
      "subscriberSystem": {
        "id": 0,
        "systemName": "string",
        "address": "string",
        "authenticationInfo": "string",
        "port": 0,
        "createdAt": "string",
        "updatedAt": "string"
      },
      "createdAt": "string",
      "updatedAt": "string"
}
```

Uri: /eventhandler/mgmt/subscriptions/{id}
Type: PUT

Update requested Subscription entry by the given id and parameters

Input JSON structure:

```
{
  "eventType": "string",
  "filterMetaData": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  },
  "matchMetaData": true,
  "notifyUri": "string",
  "sources": [
    {
      "address": "string",
      "authenticationInfo": "string",
      "port": 0,
      "systemName": "string"
    }
  ],
  "startDate": "string",
  "endDate": "string",
  "subscriberSystem": {
    "systemName": "string",
    "address": "string",
    "authenticationInfo": "string",
    "port": 0
  }
}
```

Returns the updated subscription record specified by the id and parameter.

Returned JSON structure:

```
{
      "id": 0,
      "eventType": {
        "id": 0,
        "eventTypeName": "string",
        "createdAt": "string",
        "updatedAt": "string"
      },
      "filterMetaData": {
        "additionalProp1": "string",
        "additionalProp2": "string",
        "additionalProp3": "string"
      },
      "matchMetaData": true,
      "notifyUri": "string",
      "sources": [
        {
          "id": 0,
          "systemName": "string",
          "address": "string",
          "authenticationInfo": "string",
          "port": 0,
          "createdAt": "string",
          "updatedAt": "string"
        }
      ],
      "startDate": "string",
      "endDate": "string",
      "subscriberSystem": {
        "id": 0,
        "systemName": "string",
        "address": "string",
        "authenticationInfo": "string",
        "port": 0,
        "createdAt": "string",
        "updatedAt": "string"
      },
      "createdAt": "string",
      "updatedAt": "string"
}
```

Uri: /eventhandler/mgmt/subscriptions/{id}
Type: DELETE

Removes the subscription record specified by the id path parameter.

Private Services

Not available.