

ConfigManagement HTTP/TLS/JSON

Interface Design Description

Service ID: *"configmanagement"*

Abstract

This document describes the ConfigManagement service IDD for HTTP/TLS/JSON.



ARTEMIS Innovation Pilot Project: Arrowhead
THEME [SP1-JTI-ARTEMIS-2012-AIPP4 SP1-JTI-ARTEMIS-2012-AIPP6]
[Production and Energy System Automation Intelligent-Built environment and urban infrastructure for sustainable and friendly cities]

Contents

1 Overview	3
2 Service Interfaces	4
2.1 function ManagementListConfigurations	4
2.2 function ManagementStoreConfiguration	4
2.3 function ManagementDeleteConfiguration	5
3 Information Model	6
4 References	7
5 Revision History	8
5.1 Amendments	8
5.2 Quality Assurance	8



ARROWHEAD

Document title
ConfigManagement HTTP/TLS/JSON
Date
2021-03-29

Version
1.0
Status
DRAFT
Page
3 (8)

1 Overview

This document describes the HTTP/TLS/JSON variant of the ConfigManagement Eclipse Arrowhead service. The ConfigManagement service is used to store and manage configurations and settings for application systems in a local cloud. The ConfigManagement service also enables cloud operators (sysop) to create new, delete and list configurations.

This document exists as a complement to the *ConfigManagement – Service Description* (ConfigManagement SD) document. For further details about how this service is meant to be used, please consult that document. The rest of this document describes how to realize the ConfigManagement service using HTTP [1], TLS [2] and JSON [3], both in terms of its interfaces (Section 2) and its information model (Section 3).

2 Service Interfaces

This section lists the interfaces that must be exposed by the ConfigManagement service in alphabetical order. In particular, each subsection first names the HTTP method and path used to call the interface, after which it names an abstract interface from the ConfigManagement service's SD document, as well as input and output types. All interfaces in this section respond with the HTTP status code 200 OK if called successfully, unless otherwise is stated.

2.1 GET /configuration/mgmt/config

Interface: **ManagementListConfigurations**

Output: **ConfigurationListResponse**

Called to get a complete list of all configuration objects in the database. An example is given in Listing 2.

```

1 GET /configuration/mgmt/config HTTP/1.1
2
3 RESPONSE:
4 {
5   "count": 1,
6   "data": [
7     {
8       "id": 1,
9       "systemName": "examplesystem1",
10      "contentType": "text/plain",
11      "data": "cG9ydD04ODAxCnBhdGg9ImV4YW1wbGUvcGF0aCIKaXA9IjEwLjAuMC4yMyI=",
12      "createdAt": "2021-01-26 22:23:13",
13      "updatedAt": "2021-01-26 22:23:13"
14    }
15  ]
16 }
```

Listing 1: A **ManagementListConfigurations** invocation with an JSON-encoded response.

Code	Type	Description
200	OK	No error
400	BAD REQUEST	Bad input
401	UNAUTHORIZED	No valid authorization
500	INTERNAL SERVER ERROR	Database error etc

Table 1: ManagementListConfigurations responses

2.2 PUT /configuration/mgmt/config/\$systemName

Interface: **ManagementStoreConfiguration**

Input: **ConfigurationRequest**

Output: **ConfigurationResponse**

Called to store configuration data. Only the local cloud operator (sysop) can perform this action. An example is given in Listing 2.

```

1 PUT /configuration/mgmt/config/examplesystem1 HTTP/1.1
2
3 REQUEST:
4 {
5   "systemName": "examplesystem1",
6   "contentType": "text/plain",
7   "data": "cG9ydD04ODAxCnBhdGg9ImV4YW1wbGUvcGF0aCIKaXA9IjEwLjAuMC4yMyI="
8 }
```

```

9
10 RESPONSE:
11 {
12   "id": 1,
13   "systemName": "examplesystem1",
14   "contentType": "text/plain",
15   "data": "cG9ydD04ODAxCnBhdGg9ImV4YW1wbGUvcGF0aCIKaXA9IjEwLjAuMC4yMyI=",
16   "createdAt": "2021-01-26 22:23:13",
17   "updatedAt": "2021-01-26 22:23:13"
18 }

```

Listing 2: A [ManagementStoreConfiguration](#) Response with JSON-encoded data.

Code	Type	Description
200	OK	No error
400	BAD REQUEST	Bad input
401	UNAUTHORIZED	No valid authorization
404	NOT FOUND	No such system/service combination
500	INTERNAL SERVER ERROR	Database error etc

Table 2: ManagementStoreConfiguration responses

2.3 DELETE /configuration/mgmt/config/\$systemName

Interface: [ManagementDeleteConfiguration](#)

Output: [ConfigurationResponse](#)

Called to delete a configuration object in the database. An example is given in Listing 3. The deleted entry is also returned.

```

1 DELETE /configuration/mgmt/config/examplesystem1 HTTP/1.1
2
3 RESPONSE:
4 {
5   "id": 1,
6   "systemName": "examplesystem1",
7   "contentType": "text/plain",
8   "data": "cG9ydD04ODAxCnBhdGg9ImV4YW1wbGUvcGF0aCIKaXA9IjEwLjAuMC4yMyI=",
9   "createdAt": "2021-01-26 22:23:13",
10  "updatedAt": "2021-01-26 22:23:13"
11 }

```

Listing 3: A [ManagementDeleteConfiguration](#) invocation with an JSON-encoded response.

Code	Type	Description
200	OK	No error
400	BAD REQUEST	Bad input
401	UNAUTHORIZED	No valid authorization
500	INTERNAL SERVER ERROR	Database error etc

Table 3: ManagementDeleteConfiguration responses



ARROWHEAD

Document title
ConfigManagement HTTP/TLS/JSON
Date
2021-03-29

Version
1.0
Status
DRAFT
Page
6 (8)

3 Information Model

The ConfigManagement service does not restrict the information model. It supports any type of base64-encoded data to be sent encapsulated in an JSON object. This enables Black-box / binary data configuration files to be stored, and further downloaded using the RawConfig service.



ARROWHEAD

Document title
ConfigManagement HTTP/TLS/JSON
Date
2021-03-29

Version
1.0
Status
DRAFT
Page
7 (8)

4 References

- [1] R. Fielding and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing," RFC 7230, 2018, RFC Editor. [Online]. Available: <https://doi.org/10.17487/RFC7230>
- [2] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, 2018, RFC Editor. [Online]. Available: <https://doi.org/10.17487/RFC8446>
- [3] T. Bray, "The JavaScript Object Notation (JSON) Data Interchange Format," RFC 7159, 2014, RFC Editor. [Online]. Available: <https://doi.org/10.17487/RFC7159>



ARROWHEAD

Document title
ConfigManagement HTTP/TLS/JSON
Date
2021-03-29

Version
1.0
Status
DRAFT
Page
8 (8)

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	2021-01-15	0.1	Initial	Jens Eliasson
2	2021-01-25	0.5	Text update	Jens Eliasson
3	2021-02-06	0.8	Data models	Jens Eliasson
4	2021-02-09	0.9	Updated service interfaces	Jens Eliasson
4	2021-03-28	1.0	Final version	Jens Eliasson

5.2 Quality Assurance

No.	Date	Version	Approved by
1			