query-by-system

Service Description

Abstract
This document provides service description for the **query-by-system** service.

Contents

1 Overview

This document describes the **query-by-system** service, which enables the systems to query for specific system details based on system parameters. Example of this interaction is a dedicated core system that needs the available information for a specific system.

The rest of this document is organized as follows. In Section 2, we describe the abstract message functions provided by the service. In Section 3, we end the document by presenting the data types used by the mentioned functions.

1.1 How This Service Is Meant to Be Used

The given core system is required to submit the system parameters.

1.2 Important Delimitations

System name can contain maximum 63 character of letters (english alphabet), numbers and dash (-), and have to start with a letter (also cannot end with dash).

1.3 Access policy

Available only for the following core systems: *Orchestrator, Workflow Choreographer*

2 Service Interface

This section describes the interfaces to the service. The **query-by-system** service is used to looking for a specific system. The various parameters are representing the necessary system input information. In particular, each subsection names an interface, an input type and an output type, in that order. The input type is named inside parentheses, while the output type is preceded by a colon. Input and output types are only denoted when accepted or returned, respectively, by the interface in question. All abstract data types named in this section are defined in Section 3.

The following interfaces are available.

2.1 interface HTTP/TLS/JSON (SystemRequest) : SystemResponse

Profile type	Type	Version
Transfer protocol	HTTP	1.1
Data encryption	TLS	1.3
Encoding	JSON	RFC 8259 [1]
Compression	N/A	-

Table 1: HTTP/TLS/JSON communication details.

3 Information Model

Here, all data objects that can be part of the **query-by-system** service provides to the hosting System are listed in alphabetic order. Note that each subsection, which describes one type of object, begins with the *struct* keyword, which is used to denote a collection of named fields, each with its own data type. As a complement to the explicitly defined types in this section, there is also a list of implicit primitive types in Section 3.4, which are used to represent things like hashes and identifiers.

3.1 struct SystemRequest

Field	Type	Mandatory	Description
address	Address	yes	Network address of the system.
authenticationInfo	String	no	X.509 public key of the system.
metadata	Metadata	no	Additional information about the system.
port	PortNumber	yes	Port of the system.
systemName	Name	yes	Name of the system.

3.2 struct Metadata

An Object which maps String key-value pairs.

3.3 struct SystemResponse

Field	Type	Description
address	Address	Network address of the system.
authenticationInfo	String	X.509 public key of the system.
createdAt	DateTime	System instance record was created at this UTC time-stamp.
id	Number	Identifier of the system instance.
metadata	Metadata	Additional information about the system.
port	PortNumber	Port of the system.
systemName	Name	Name of the system.
updatedAt	DateTime	System instance record was modified at this UTC time-stamp.

3.4 Primitives

Types and structures mentioned throughout this document that are assumed to be available to implementations of this service. The concrete interpretations of each of these types and structures must be provided by any IDD document claiming to implement this service.

Type	Description
Address	A string representation of the address
DateTime	Pinpoints a specific moment in time.
Object	Set of primitives and possible further objects.
Name	A string identifier that is intended to be both human and machine-readable.
Number	Decimal number.
PortNumber	A Number between 0 and 65535.
String	A chain of characters.

4 References

[1] T. Bray, “The JavaScript Object Notation (JSON) Data Interchange Format,” RFC 8259, Dec. 2017. [Online]. Available: https://rfc-editor.org/rfc/rfc8259.txt

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	YYYY-MM-DD	4.6.0		Xxx Yyy

5.2 Quality Assurance

No.	Date	Version	Approved by
1	YYYY-MM-DD	4.6.0	