create-digital-twin

Service Description

Abstract
This document provides service description for the **create-digital-twin** service.

Contents

1 Overview

This document describes the **create-digital-twin** service, which makes it possible to create a digital twin. The service will register all services defined in the request and create the the digital twin, furthermore the digital twin system will be saved in the digital twin hub database.

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 system operator should use the **create-digital-twin** service when they want to create a new digital twin system in the local cloud. They must provide a way the digital twin can connect to a physical twin and they must define which endpoints they want to extend an reflect from the physical twin.

1.2 Important Delimitations

The registration data must meet the following criteria:

- connectionType specifies what protocol (like CoAP, MQTT or OPC-UA) the digital twin will use to connect to the physical twin. The desired connection type must have an implementation in the digital twin hub, currently only a simplistic CoAP connection is implemented and it is defined by `simple-CoAP`.
- sensorEndpointMode currently has the modes `IMMEDIATE_RETRIEVAL` or `INTERVAL_RETRIEVAL`. `IMMEDIATE_RETRIEVAL` mode will retrieve the sensor data from the physical twin immediately when a sensor request is received. In the `INTERVAL_RETRIEVAL` mode, the digital twin will query the physical twin automatically after the specified interval time and store it in a database, when a sensor request is received the latest value in the database will be returned.

1.3 Access policy

Available only for system operators, it is necessary that a sysop certificate is provided.

2 Service Interface

This section describes the interfaces to the service. The **create-digital-twin** service is used to create a digital twin system. 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 (DigitalTwinRequest) : DigitalTwinResponse

Profile type	Type	Version
Transfer protocol	HTTP	1.1
Data encryption	TLS	1.3
Encoding	JSON	-
Method	POST	-

Table 1: HTTP/TLS/JSON communication details.

3 Information Model

Here, all data objects that can be part of the **create-digital-twin** 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 3.7, which are used to represent things like hashes and identifiers.

3.1 struct DigitalTwinRequest

Field	Type	Mandatory	Description
controlCommands	List<ControlCommand>	no	Define control endpoints
physicalTwinConnection	PhysicalTwinConnection	yes	Connection settings to the physical twin.
sensedProperties	List<SensedProperty>	no	Define sensor endpoints

3.2 struct PhysicalTwinConnection

Field	Type	Mandatory	Description
connectionModel	ConnectionModel	yes	Defines parameters with how the digital twin connects to the physical twin.
connectionType	String	yes	Defines the type of connection that the digital twin uses to connect to the digital twin

3.3 struct ConnectionModel

Field	Type	Mandatory	Description
address	Address	yes	Network address.
port	PortNumber	yes	Network port.

3.4 struct ControlCommand

Field	Type	Mandatory	Description
serviceDefinition	String	yes	Definition of the service that will be registered that correspondes to the created endpoint.
serviceUri	String	yes	Uri path for the created endpoint.

3.5 struct SensedProperty

Field	Type	Mandatory	Description
intervalTime	String	no	If the "INTERVAL_RETRIEVAL" sensor type is used then this defines the interval in seconds.
sensorEndpointMode	String	yes	Defines the type of sensor endpoint that will be used by the digital twin.
serviceDefinition	String	yes	Definition of the service that will be registered that correspondes to the created endpoint.
serviceUri	String	yes	Uri path for the created endpoint.

3.6 struct DigitalTwinResponse

Field	Type	Description
address	Address	he created digital twins network address.
authenticationInfo	String	X.509 public key of the digital twin system.
port	PortNumber	The created digital twins network port.
systemName	String	Name of the created digital twin system.

3.7 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
List<A>	An *array* of a known number of items, each having type A.
PortNumber	A Number between 0 and 65535.
String	A chain of characters.

4 References

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	YYYY-MM-DD	4.6.1		Xxx Yyy

5.2 Quality Assurance

No.	Date	Version	Approved by
1	YYYY-MM-DD	4.6.1	