query-multi

Service Description

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

Contents

1 Overview

This document describes the **query-multi** service, which enables autonomous service look up for multiple services by one request, therefore it is an integral part of the implementation of service discovery requirements in Service Registry Mandatory Core System. Example of this interaction is a core system that needs the available information for services that fulfill certain requirements. The look up process itself is exactly the same as in case of the **query** service.

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 a list of service query forms, in which the requirements for each systems are defined.

1.2 Important Delimitations

The query data must meet the following criteria:

- Service definition can contain maximum 63 character of letters (english alphabet), numbers and dash (-), and has to start with a letter (also cannot ends with dash).
- Interface names have to follow the `Protocol-SecurityType-MimeType` format.
- Security types could be only `NOT_SECURE`, `CERTIFICATE` or `TOKEN` .

1.3 Access policy

Available only for the following core systems: *Authorization, Orchestrator, Certificate Authority, Workflow Choreographer, Device Registry, Event Handler, HawkBit Configuration Manager, MSCV, Plant Description Engine, QoS-Monitor, System Registry*.

2 Service Interface

This section describes the interfaces to the service. The **query-multi** service is used to looking for multiple services. A service could contain various metadata as well as a physical endpoint. The various parameters are representing the necessary system and service 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 (ServiceQueryFormList) : ServiceQueryResultList

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-multi** 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 3.10, which are used to represent things like hashes and identifiers.

3.1 struct ServiceQueryFormList

Field	Type	Mandatory	Description
forms	List<ServiceQueryForm>	yes	List of service requirements form.

3.2 struct ServiceQueryForm

Field	Type	Mandatory	Description
interfaceRequirements	List<Interface>	no	Names of the required interfaces. If specified at least one of the interfaces must match for having result(s).
maxVersionRequirement	Number	no	Required maximum version of the service. If specified version must be equals or lower for having result(s). Ignored if *versionRequirement* is specified.
metadataRequirements	Metadata	no	Service metadata requirements. If specified the whole content of the map must match for having result(s).
minVersionRequirement	Number	no	Required minimum version of the service. If specified version must be equals for higher or having result(s). Ignored if *versionRequirement* is specified.
pingProviders	Boolean	no	Whether or not the provider should be pinged. If *true* only the responding providers will comply.
securityRequirements	List<SecureType>	no	Types of the required security levels. If specified at least one of the types must match for having result(s).
serviceDefinitionRequirement	Name	yes	Identifier of the service.
versionRequirement	Number	no	Required version of the service. If specified version must match for having result(s).

3.3 struct Metadata

An Object which maps String key-value pairs.

3.4 struct ServiceQueryResultList

Field	Type	Description
results	List<ServiceQueryResult>	List of service query results.

3.5 struct ServiceQueryResult

Field	Type	Description
serviceQueryData	List<ServiceRegistryResponse>	List of service instance results.
unfilteredHits	Number	Number of service instances without the specified filters.

3.6 struct ServiceRegistryResponse

Field	Type	Description
createdAt	DateTime	Service instance record was created at this UTC timestamp.
endOfValidity	DateTime	Service is available until this UTC timestamp.
id	Number	Identifier of the service instance.
interfaces	List<InterfaceRecord>	List of interfaces the service supports.
provider	SystemRecord	Descriptor of the provider system record.
secure	SecureType	Type of security the service uses.
serviceDefinitionResponse	ServiceDefinitionRecord	Descriptor of the service definition record.
serviceUri	String	Path of the service on the provider.
metadata	Metadata	Service metadata.
updatedAt	DateTime	Service instance record was modified at this UTC timestamp.
version	Version	Version of the service.

3.7 struct InterfaceRecord

Field	Type	Description
createdAt	DateTime	Interface instance record was created at this UTC timestamp.
id	Number	Identifier of the interface instance.
interfaceName	Interface	Specified name of the interface.
updatedAt	DateTime	Interface instance record was modified at this UTC timestamp.

3.8 struct SystemRecord

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 timestamp.
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 timestamp.

3.9 struct ServiceDefinitionRecord

Field	Type	Description
createdAt	DateTime	Service definition instance record was created at this UTC timestamp.
id	Number	Identifier of the service definition instance.
serviceDefinition	Name	Name of the service definition.
updatedAt	DateTime	Service definition instance record was modified at this UTC timestamp.

3.10 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.
Boolean	One out of `true` or `false`.
DateTime	Pinpoints a specific moment in time.
Interface	Any suitable type chosen by the implementor of service
List<A>	An *array* of a known number of items, each having type A.
Object	Set of primitives and possible further objects.
PortNumber	A Number between 0 and 65535.
Name	A string identifier that is intended to be both human and machine-readable.
Number	Decimal number
SecureType	Any suitable type chosen by the implementor of service
String	A chain of characters.
Version	Specifies a service version.

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	