serviceRegistryManagement

Service Description

Abstract
This document provides service description for the **serviceRegistryManagement** service.

Contents

1 Overview

This document describes the **serviceRegistryManagement** service, which enables systems (with operator role or proper permissions) to handle (register, update, revoke, lookup) devices, systems, service instances, service definitions, and interface templates in bulk. An example of this interaction is when an operator uses the Management Tool to register interface templates, systems, and service instances manually.

The **serviceRegistryManagement** service contains the following operations:

- *device-query* lists the devices that match the filtering requirements;
- *device-create* registers the specified devices;
- *device-updates* updates the specified existing devices;
- *device-remove* revokes the specified devices;
- *system-query* lists the systems that match the filtering requirements;
- *system-create* registers the specified systems;
- *system-update* updates the specified existing systems;
- *system-remove* revokes the specified systems;
- *service-definition-query* lists the service definitions;
- *service-definition-create* registers the specified service definitions;
- *service-definition-remove* revokes the specified service definitions;
- *service-query* lists the service instances that match the filtering requirements;
- *service-create* registers the specified service instances;
- *service-update* updates the specified existing service instances;
- *service-remove* revokes the specified service instances;
- *interface-template-query* lists the interface templates that match the filtering requirements;
- *interface-template-create* registers the specified interface templates;
- *interface-template-remove* revokes the specified interface templates;

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

1.1 How This Service Is Meant to Be Used

The service's purpose is to handle the various entities of the ServiceRegistry centrally and in bulk. Such entities are devices, systems, service definitions, service instances and interface templates. This approach makes possible that the individual application systems do not have to do anything to be able to work within the Local Cloud.

Application systems should not use this service; only operators (via the Management Tool, for example) or dedicated Support systems.

1.2 Important Delimitations

The requester has to identify itself to use any of the operations.

1.3 Access policy

The service is only available for operators, dedicated Support systems and those who have the proper authorization rights to consume it.

2 Service Operations

This section describes the abstract signatures of each operation of the service. In particular, each subsection names an operation, an input type, and one or two output types (unsuccessful operations can return different structure), in that order. The input type is named inside parentheses, while the output type is preceded by a colon. If the operation has two output types, they are separated by a slash. Input and output types are only denoted when accepted or returned, respectively, by the operation in question. All abstract data types named in this section are defined in Section 3.

2.1 operation device-query (DeviceQueryRequest) : DeviceListResponse / ErrorResponse

Operation *device-query* lists the devices that match the filtering requirements. The query data must meet the following criteria:

- The operation returns results in pages. There are default page data settings, but the requester can provide a custom specification.
- If page number is specified, the page size must be specified as well and vice versa.
- In some Local Clouds there is a maximum page size.
- If a filter expects a list, there is an OR relation between the elements of the filter.
- There is an AND relation between different kind of filters.

2.2 operation device-create (DeviceListRequest) : DeviceListResponse / ErrorResponse

Operation *device-create* registers the specified devices. The creation data must meet the following criteria:

- Device names are case sensitive, must follow the UPPER_SNAKE_CASE naming convention and have to be unique within the Local Cloud.
- Device names can contain maximum 63 character of uppercase letters (English alphabet), numbers and underscore (_), and have to start with a letter (also cannot end with underscore).
- Device has to have at least one address. It is recommended to use MAC address and/or IP address.
- Keys in the metadata structure can not contain dot (.) character.
- Multiple devices can be created at once, but it is forbidden to specify devices with the same name.
- It is forbidden to create a device that is already registered into the Local Cloud.

2.3 operation device-update (DeviceListRequest) : DeviceListResponse / ErrorResponse

Operation *device-updates* updates the specified existing devices. The update data must meet the following criteria:

- Device names are case sensitive, must follow the UPPER_SNAKE_CASE naming convention and have to be unique within the Local Cloud.
- Device names can contain maximum 63 character of uppercase letters (English alphabet), numbers and underscore (_), and have to start with a letter (also cannot end with underscore).
- Device has to have at least one address. It is recommended to use MAC address and/or IP address.
- Keys in the metadata structure can not contain dot (.) character.
- Multiple devices can be updated at once, but it is forbidden to update devices with the same name.
- It is not possible to update a device that is not registered into the Local Cloud.

2.4 operation device-remove (DeviceRemoveRequest) : OperationStatus / ErrorResponse

Operation *device-remove* revokes the specified devices. The input operation data must meet the following criteria:

- The operation can only revoke devices without any system assignment.

2.5 operation system-query (SystemQueryRequest) : SystemListResponse / ErrorResponse

Operation *system-query* lists the systems that match the filtering requirements. The query data must meet the following criteria:

- The operation returns results in pages. There are default page data settings, but the requester can provide a custom specification.
- If page number is specified, the page size must be specified as well and vice versa.
- In some Local Clouds there is a maximum page size.
- If a filter expects a list, there is an OR relation between the elements of the filter.
- There is an AND relation between different kind of filters.

2.6 operation system-create (SystemListRequest) : SystemListResponse / ErrorResponse

Operation *system-create* registers the specified systems. The creation data must meet the following criteria:

- System names are case sensitive, must follow the PascalCase naming convention and have to be unique within the Local Cloud.
- System names can contain maximum 63 character of letters (English alphabet), and numbers, and have to start with a letter.
- System has to have at least one address (including those that come from the referenced device, if any). It is recommended to use IP address.
- Keys in the metadata structure can not contain dot (.) character.
- Device, if specified, has to be present in the Local Cloud.
- Multiple systems can be created at once, but it is forbidden to specify systems with the same name.
- It is forbidden to create a system that is already registered into the Local Cloud.

2.7 operation system-update (SystemListRequest) : SystemListResponse / ErrorResponse

Operation *system-update* updates the specified existing systems. The update data must meet the following criteria:

- System names are case sensitive, must follow the PascalCase naming convention and have to be unique within the Local Cloud.
- System names can contain maximum 63 character of letters (English alphabet), and numbers, and have to start with a letter.
- System has to have at least one address (including those that come from the referenced device, if any). It is recommended to use IP address.
- Keys in the metadata structure can not contain dot (.) character.
- Device, if specified, has to be present in the Local Cloud.
- Multiple systems can be updated at once, but it is forbidden to update systems with the same name.
- It is not possible to update a system that is not registered into the Local Cloud.

2.8 operation system-remove (SystemRemoveRequest) : OperationStatus / ErrorResponse

Operation *system-remove* revokes the specified systems.

2.9 operation service-definition-query (PageRequest) : ServiceDefinitionListResponse / ErrorResponse

Operation *service-definition-query* lists the service definitions. The query data must meet the following criteria:

- The operation returns results in pages. There are default page data settings, but the requester can provide a custom specification.
- If page number is specified, the page size must be specified as well and vice versa.
- In some Local Clouds there is a maximum page size.

2.10 operation service-definition-create (ServiceDefinitionListRequest) : ServiceDefinitionListResponse / ErrorResponse

Operation *service-definition-create* registers the specified service definitions. The creation data must meet the following criteria:

- Service definition names are case sensitive, must follow the camelCase naming convention and have to be unique within the Local Cloud.
- Service definition names can contain maximum 63 character of letters (English alphabet) and numbers, and have to start with a letter.
- Multiple service definitions can be created at once, but it is forbidden to specify service definitions with the same name.
- It is forbidden to create a service definition that is already registered into the Local Cloud.

2.11 operation service-definition-remove (ServiceDefinitionRemoveRequest) : OperationStatus / ErrorResponse

Operation *service-definition-remove* revokes the specified service definitions.

2.12 operation service-query (ServiceQueryRequest) : ServiceListResponse / ErrorResponse

Operation *service-query* lists the service instances that match the filtering requirements. The query data must meet the following criteria:

- The operation returns results in pages. There are default page data settings, but the requester can provide a custom specification.
- If page number is specified, the page size must be specified as well and vice versa.
- In some Local Clouds there is a maximum page size.
- If a filter expects a list, there is an OR relation between the elements of the filter.

- There is an AND relation between different kind of filters.
- To use this operation, an application system must specify at least one service instance id OR one provider name OR one service definition.

2.13 operation service-create (ServiceCreateListRequest) : ServiceListResponse / ErrorResponse

Operation *service-create* registers the specified service instances. The creation data must meet the following criteria:

- The provider systems have to be present in the Local Cloud.
- Service definition names are case sensitive, must follow the camelCase naming convention and have to be unique within the Local Cloud.
- Service definition names can contain maximum 63 character of letters (English alphabet) and numbers, and have to start with a letter.
- If an expiration date is specified, it cannot point to a past date.
- Keys in the metadata structure can not contain dot (.) character.
- For every instance, at least one interface must be specified with its template name, the used policy and all the necessary template-specific properties.
- If a specified interface template is not present in the Local Cloud, a protocol also has to be defined. In some Local Clouds using previously unknown interface templates during the service instance creation may be forbidden and thus rejected.
- Multiple service instances can be created at once, but it is forbidden to specify service instances with the same provider, service definition and version.
- Service instances can be created multiple times, the appropriate old service instances are discarded.

2.14 operation service-update (ServiceUpdateListRequest) : ServiceListResponse / ErrorResponse

Operation *service-update* updates the specified existing service instances. The update data must meet the following criteria:

- Only expiration date, metadata and interfaces can be updated with this operation.
- If an expiration date is specified, it cannot point to a past date.
- Keys in the metadata structure can not contain dot (.) character.
- For every instance, at least one interface must be specified with its template name, the used policy and all the necessary template-specific properties.
- If a specified interface template is not present in the Local Cloud, a protocol also has to be defined. In some Local Clouds using previously unknown interface templates during the service instance creation may be forbidden and thus rejected.

- Multiple service instances can be updated at once, but it is forbidden to update service instances with the same service instance ids.
- It is not possible to update a service instance that is not registered into the Local Cloud.

2.15 operation service-remove (ServiceRemoveRequest) : OperationStatus / ErrorResponse

Operation *service-remove* revokes the specified service instances.

2.16 operation interface-template-query (InterfaceTemplateQueryRequest) : InterfaceTemplateListResponse / ErrorResponse

Operation *interface-template-query* lists the interface templates that match the filtering requirements. The query data must meet the following criteria:

- The operation returns results in pages. There are default page data settings, but the requester can provide a custom specification.
- If page number is specified, the page size must be specified as well and vice versa.
- In some Local Clouds there is a maximum page size.
- If a filter expects a list, there is an OR relation between the elements of the filter.
- There is an AND relation between different kind of filters.

2.17 operation interface-template-create (InterfaceTemplateListRequest) : InterfaceTemplateListResponse / ErrorResponse

Operation *interface-template-create* registers the specified interface templates. The creation data must meet the following criteria:

- Interface template names are case sensitive, must follow the snake_case naming convention and have to be unique within the Local Cloud.
- Interface template names can contain maximum 63 character of lowercase letters (English alphabet), numbers and underscore (_), and have to start with a letter (also cannot end with underscore).
- Protocol names are case insensitive and can contain maximum 63 characters.
- Property names can contain maximum 63 characters and can not contain dot (.) character.
- It is forbidden to specify properties with the same name for the same interface templates.
- Validator is optional, but it is forbidden to specify validator parameters without a validator.
- Multiple interface templates can be created at once, but it is forbidden to specify interface templates with the same name.
- It is forbidden to create an interface template that is already registered into the Local Cloud.

2.18 operation interface-template-remove (InterfaceTemplateRemoveRequest) : OperationStatus / ErrorResponse

Operation *interface-template-remove* revokes the specified interface templates.

3 Information Model

Here, all data objects that can be part of the **serviceRegistryManagement** service are listed and must be respected by the hosting System. 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 3.41, which are used to represent things like hashes and identifiers.

3.1 struct DeviceQueryRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
pageNumber	Number	no (yes)	The number of the requested page. It is mandatory, if page size is specified.
pageSize	Number	no (yes)	The number of entries on the requested page. It is mandatory, if page number is specified.
pageSortField	String	no	The identifier of the field which must be used to sort the entries.
pageDirection	Direction	no	The direction of the sorting.
deviceNames	List<DeviceName>	no	Requester is looking for devices with any of the specified names.
addresses	List<Address>	no	Requester is looking for devices with any of the specified addresses.
addressType	AddressType	no	Requester is looking for devices with the specified type of address.
metadataRequirementsList	List<MetadataRequirements>	no	Requester is looking for devices that are matching any of the specified metadata requirements.

3.2 struct Identity

An Object which describes the identity of a system. It also contains whether the identified system has higher level administrative rights.

3.3 struct MetadataRequirements

A special Object which maps String keys to Object, primitive or list values, where

- Keys can be paths (or multi-level keys) which access a specific value in a Metadata structure, where parts of the path are delimited with dot character (e.g. in case of ”key.subkey” path we are looking for the key

named ”key” in the metadata, which is associated with an embedded object and in this object we are looking for the key named ”subkey”).

- Values are special Objects with two fields: an operation (e.g. less than) and an actual value (e.g. a number). A metadata is matching a requirement if the specified operation returns true using the metadata value referenced by a key path as first and the actual value as second operands.
- Alternatively, values can be ordinary primitives, lists or Objects. In this case the operation is equals by default.

3.4 struct DeviceListResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
entries	List<DeviceResult>	A page of devices.
count	Number	Total number of devices that match the filters.

3.5 struct DeviceResult

Field	Type	Description
name	DeviceName	Unique identifier of the device.
metadata	Metadata	Additional information about the device.
addresses	List<AddressDescriptor>	Different kind of addresses of the device.
createdAt	DateTime	Device was registered at this timestamp.
updatedAt	DateTime	Device was modified at this timestamp.

3.6 struct Metadata

An Object which maps String keys to primitive, Object or list values.

3.7 struct AddressDescriptor

Field	Type	Description
type	AddressType	Type of the address.
address	Address	Address.

3.8 struct ErrorResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
errorMessage	String	Description of the error.
errorCode	Number	Numerical code of the error.
type	ErrorType	Type of the error.
origin	String	Origin of the error.

3.9 struct DeviceListRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
devices	List<DeviceRequest>	yes	A list of devices.

3.10 struct DeviceRequest

Field	Type	Mandatory	Description
name	DeviceName	yes	Unique identifier of the device.
metadata	Metadata	no	Additional information about the device.
addresses	List<Address>	yes	Different kind of addresses of the device.

3.11 struct DeviceRemoveRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
names	List<DeviceName>	yes	Names of the devices that need to be removed.

3.12 struct SystemQueryRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
verbose	Boolean	no	If true detailed device information also returns.
pageNumber	Number	no (yes)	The number of the requested page. It is mandatory, if page size is specified.

pageSize	Number	no (yes)	The number of entries on the requested page. It is mandatory, if page number is specified.
pageSortField	String	no	The identifier of the field which must be used to sort the entries.
pageDirection	Direction	no	The direction of the sorting.
systemNames	List<SystemName>	no	Requester is looking for systems with any of the specified names.
addresses	List<Address>	no	Requester is looking for systems with any of the specified addresses.
addressType	AddressType	no	Requester is looking for systems with the specified type of address.
metadataRequirementsList	List<MetadataRequirements>	no	Requester is looking for systems that are matching any of the specified metadata requirements.
versions	List<Version>	no	Requester is looking for systems with any of the specified versions.
deviceNames	List<DeviceName>	no	Requester is looking for systems that are running on any of the specified devices.

3.13 struct SystemListResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
entries	List<SystemResult>	A page of systems.
count	Number	Total number of systems that match the filters.

3.14 struct SystemResult

Field	Type	Description
name	SystemName	Unique identifier of the system.
metadata	Metadata	Additional information about the system.
version	Version	Version of the system.
addresses	List<AddressDescriptor>	Different kind of addresses of the system.
device	DeviceResult	Information about the device on which the system is running.
createdAt	DateTime	System was registered at this timestamp.
updatedAt	DateTime	System was modified at this timestamp.

3.15 struct SystemListRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
systems	List<SystemRequest>	yes	A list of systems.

3.16 struct SystemRequest

Field	Type	Mandatory	Description
name	SystemName	yes	The name of the system.
metadata	Metadata	no	Additional information about the system.
version	Version	no	Version of the system.
addresses	List<Address>	yes	Different kind of addresses of the system.
deviceName	DeviceName	no	Unique identifier of the device on which the system is running.

3.17 struct SystemRemoveRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
names	List<SystemName>	yes	Names of the systems that need to be removed.

3.18 struct PageRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
pageNumber	Number	no (yes)	The number of the requested page. It is mandatory, if page size is specified.
pageSize	Number	no (yes)	The number of entries on the requested page. It is mandatory, if page number is specified.
pageSortField	String	no	The identifier of the field which must be used to sort the entries.
pageDirection	Direction	no	The direction of the sorting.

3.19 struct ServiceDefinitionListResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
entries	List<ServiceDefinitionResult>	A page of service definitions.
count	Number	Total number of service definitions.

3.20 struct ServiceDefinitionResult

Field	Type	Description
name	ServiceName	Unique identifier of the service definition.
createdAt	DateTime	Service definition was registered at this timestamp.
updatedAt	DateTime	Service definition was modified at this timestamp.

3.21 struct ServiceDefinitionListRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
serviceDefinitionNames	List<ServiceName>	yes	A list of service definition names.

3.22 struct ServiceDefinitionRemoveRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
names	List<ServiceName>	yes	Names of the service definitions that need to be removed.

3.23 struct ServiceQueryRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
verbose	Boolean	no	If true detailed system and device information also returns.
pageNumber	Number	no (yes)	The number of the requested page. It is mandatory, if page size is specified.

pageSize	Number	no (yes)	The number of entries on the requested page. It is mandatory, if page number is specified.
pageSortField	String	no	The identifier of the field which must be used to sort the entries.
pageDirection	Direction	no	The direction of the sorting.
instanceIds	List<ServiceInstanceID>	no (yes)	Requester is looking for service instances with any of the specified identifiers. Mandatory if no providerNames nor serviceDefinitionNames are specified.
providerNames	List<SystemName>	no (yes)	Requester is looking for service instances that are provided by any of the specified systems. Mandatory if no serviceInstanceIds nor serviceDefinitionNames are specified.
serviceDefinitionNames	List<ServiceName>	no (yes)	Requester is looking for service instances with any of the specified service definition names. Mandatory if no serviceInstanceIds nor providerNames are specified.
versions	List<Version>	no	Requester is looking for service instances with any of the specified versions.
alivesAt	DateTime	no	Requester is looking for service instances that will be available at the specified moment of the future.
metadataRequirementsList	List<MetadataRequirements>	no	Requester is looking for service instances that are matching any of the specified metadata requirements.
addressTypes	List<AddressType>	no	Requester is looking for service instances with interfaces whose access addresses are matching any of these types

interfaceTemplateNames	List<InterfaceName>	no	Requester is looking for service instances with any of the specified interface template names.
interfacePropertyRequirementsList	List<MetadataRequirements>	no	Requester is looking for service instances with interfaces that are matching any of the specified properties requirements.
policies	List<SecurityPolicy>	no	Requester is looking for service instances with any of the specified security policies.

3.24 struct ServiceListResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
entries	List<ServiceResult>	A page of service instances.
count	Number	Total number of service instances that match the filters.

3.25 struct ServiceResult

Field	Type	Description
instanceId	ServiceInstanceID	Unique identifier of the service instance.
provider	SystemResult	Information about the service instance provider system.
serviceDefinition	ServiceDefinitionResult	Information about the service definition.
version	Version	Version of the service instance.
expiresAt	DateTime	The moment of the future from which the service instance will not be available.
metadata	Metadata	Additional information about the service instance.
interfaces	List<ServiceInterfaceDescriptor>	Available access interfaces of the service instance.
createdAt	DateTime	Service instance was registered at this timestamp.
updatedAt	DateTime	Service instance was modified at this timestamp.

3.26 struct ServiceInterfaceDescriptor

Field	Type	Description
templateName	InterfaceName	The name of the interface template that describes the interface structure.
protocol	Protocol	The communication protocol of the interface.
policy	SecurityPolicy	The security of the interface.
properties	Metadata	Interface template-specific data.

3.27 struct ServiceCreateListRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
instances	List<ServiceRequest>	yes	A list of service instances.

3.28 struct ServiceRequest

Field	Type	Mandatory	Description
systemName	SystemName	yes	The provider of the instance.
serviceDefinitionName	ServiceName	yes	The service definition of the instance.
version	Version	no	Version of the service instance.
expiresAt	DateTime	no	The moment of the future from which the service instance will not be available.
metadata	Metadata	no	Additional information about the service instance.
interfaces	List<ServiceInterfaceRequest>	yes	Available access interfaces of the service instance.

3.29 struct ServiceInterfaceRequest

Field	Type	Mandatory	Description
templateName	InterfaceName	yes	The name of the interface template that describes the interface structure.
protocol	Protocol	no (yes)	The communication protocol of the interface. Only mandatory if the interface template is not previously known in the Local Cloud.
policy	SecurityPolicy	yes	The security of the interface.
properties	Metadata	yes	Interface template-specific data.

3.30 struct ServiceUpdateListRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
instances	List<ServiceUpdateRequest>	yes	A list of service instances.

3.31 struct ServiceUpdateRequest

Field	Type	Mandatory	Description
instanceId	ServiceInstanceID	yes	The unique identifier of the service instance.
expiresAt	DateTime	no	The moment of the future from which the service instance will not be available.
metadata	Metadata	no	Additional information about the service instance.
interfaces	List<ServiceInterfaceRequest>	yes	Available access interfaces of the service instance.

3.32 struct ServiceRemoveRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
instances	List<ServiceInstanceID>	yes	Identitifers of the service instances that need to be removed.

3.33 struct InterfaceTemplateQueryRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
pageNumber	Number	no (yes)	The number of the requested page. It is mandatory, if page size is specified.
pageSize	Number	no (yes)	The number of entries on the requested page. It is mandatory, if page number is specified.
pageSortField	String	no	The identifier of the field which must be used to sort the entries.
pageDirection	Direction	no	The direction of the sorting.
templateNames	List<InterfaceName>	no	Requester is looking for interface templates with any of the specified names.
protocols	List<Protocol>	no	Requester is looking for interface templates with any of the specified protocols.

3.34 struct InterfaceTemplateListResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
entries	List<InterfaceTemplateResult>	A page of interface templates.
count	Number	Total number of interface templates that match the filters.

3.35 struct InterfaceTemplateResult

Field	Type	Description
name	InterfaceName	Unique name of the interface template.
protocol	Protocol	Protocol of the interface template.
propertyRequirements	List<InterfaceTemplatePropertyDescriptor>	Properties of the interface template.
createdAt	DateTime	Interface template was registered at this timestamp.
updatedAt	DateTime	Interface template was modified at this timestamp.

3.36 struct InterfaceTemplatePropertyDescriptor

Field	Type	Description
name	String	Name of the property.
mandatory	Boolean	True if the property is mandatory, false if optional.
validator	PropertyValidator	Name of the validator assigned to the property.
validatorParams	List<String>	Parameter values of the validator (if needed).

3.37 struct InterfaceTemplateListRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
interfaceTemplates	List<InterfaceTemplateRequest>	yes	A list of interface templates.

3.38 struct InterfaceTemplateRequest

Field	Type	Mandatory	Description
name	InterfaceName	yes	Unique name of the interface template.
protocol	Protocol	yes	Protocol of the interface template.
propertyRequirements	List<InterfaceTemplatePropertyRequest>	yes	Properties of the interface template.

3.39 struct InterfaceTemplatePropertyRequest

Field	Type	Mandatory	Description
name	String	yes	Name of the property.
mandatory	Boolean	yes	True if the property is mandatory, false if optional.
validator	PropertyValidator	no	Name of the validator assigned to the property.
validatorParams	List<String>	no	Parameter values of the validator.

3.40 struct InterfaceTemplateRemoveRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
names	List<InterfaceName>	yes	Names of the interface templates that need to be removed.

3.41 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.
AddressType	Any suitable type chosen by the implementor of service.
Boolean	One out of true or false.
DateTime	Pinpoints a specific moment in time.
DeviceName	A string identifier that is intended to be both human and machine-readable. Must follow the UPPER_SNAKE_CASE naming convention.
Direction	The direction of a sorting operation. Possible values are the representation of ascending or descending order.
ErrorType	Any suitable type chosen by the implementor of service.

InterfaceName	A string identifier of an interface descriptor. Must follow snake_case naming convention.
List<A>	An *array* of a known number of items, each having type A.
Number	Decimal number.
Object	Set of primitives and possible further objects.
OperationStatus	Logical, textual or numerical value that indicates whether an operation is a success or a failure. Multiple values can be used for success and error cases to give additional information about the nature of the result.
PropertyValidator	An identifier of any suitable validator function chosen by the implementor of service.
Protocol	A string representation of a communication protocol.
SecurityPolicy	Any suitable security policy chosen by the implementor of service.
ServiceInstanceID	A composite string identifier that is intended to be both human and machine-readable. It consists of the instance's provider name, service definition and version, each separated by a special delimiter character. Each part must follow its related naming convention.
ServiceName	A string identifier that is intended to be both human and machine-readable. Must follow camelCase naming convention.
String	A chain of characters.
SystemName	A string identifier that is intended to be both human and machine-readable. Must follow PascalCase naming convention.
Version	Specifies a service instance version. Version must follow the Semantic Versioning.

4 References

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	YYYY-MM-DD	5.0.0		Xxx Yyy

5.2 Quality Assurance

No.	Date	Version	Approved by
1	YYYY-MM-DD	5.0.0	