

# serviceOrchestrationHistoryManagement

## Service Description

### Abstract

This document provides service description for the **serviceOrchestrationHistoryManagement** service.

## Contents

<b>1 Overview</b>	<b>3</b>
1.1 How This Service Is Meant to Be Used . . . . .	3
1.2 Important Delimitations . . . . .	3
1.3 Access policy . . . . .	3
<b>2 Service Operations</b>	<b>4</b>
2.1 operation <code>query</code> . . . . .	4
<b>3 Information Model</b>	<b>5</b>
3.1 struct <code>OrchestrationHistoryQueryRequest</code> . . . . .	5
3.2 struct <code>Identity</code> . . . . .	5
3.3 struct <code>OrchestrationHistoryResponse</code> . . . . .	5
3.4 struct <code>OrchestrationJobResponse</code> . . . . .	6
3.5 struct <code>ErrorResponse</code> . . . . .	6
3.6 Primitives . . . . .	7
<b>4 References</b>	<b>8</b>
<b>5 Revision History</b>	<b>9</b>
5.1 Amendments . . . . .	9
5.2 Quality Assurance . . . . .	9

# 1 Overview

This document describes the **serviceOrchestrationHistoryManagement** service, which enables systems (with operator role or proper permissions) to query the service orchestration job details. An example of this interaction is that a higher entity (a dedicated system directly or a human operator indirectly via some tool) with management access queries the orchestration history in order to verify whether a specific orchestration process was executed properly or not.

Service orchestration job records are created and updated during every single execution of the orchestration process (see *serviceOrchestration* and *serviceOrchestrationPushManagement* service descriptions) and storing status and other kind of relevant information about the given orchestration processes.

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

- *query* lists the orchestration history records that match the filtering requirements;

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 purpose of the service is to list the stored orchestration jobs according to given filtering requirements.

The most common use case when a higher entity has triggered some push orchestration and wants to verify that all of them were executed without any issue and matching service instances were orchestrated and successfully pushed to the consumer 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 Core/Support systems and those who have the proper authorization rights to consume it.

## 2 Service Operations

This section describes the abstract signatures of each operations of the service. The **serviceOrchestrationHistoryManagement** service is used to *query* the orchestration job records. 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 **query** (**OrchestrationHistoryQueryRequest**) : **OrchestrationHistoryResponse** / **ErrorResponse**

Operation *query* lists the orchestration job records 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.

## 3 Information Model

Here, all data objects that can be part of the **serviceOrchestrationLockManagement** 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 3.6, which are used to represent things like hashes and identifiers.

### 3.1 struct **OrchestrationHistoryQueryRequest**

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.
ids	List<OrchestrationJobId>	no	Requester is looking for job records with any of the specified job identifiers.
statuses	List<OrchestrationJobStatus>	no	Requester is looking for job records with any of the specified job status.
type	OrchestrationType	no	Requester is looking for job records with the specified type.
requesterSystems	List<SystemName>	no	Requester is looking for job records with any of the specified requester system.
targetSystems	List<SystemName>	no	Requester is looking for job records with any of the specified target system.
serviceDefinitions	List<ServiceName>	no	Requester is looking for job records with any of the specified service definition.
subscriptionIds	List<OrchestrationSubscriptionId>	no	Requester is looking for job records with any of the specified subscription identifiers.

### 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 **OrchestrationHistoryResponse**

Field	Type	Description
status	OperationStatus	Status of the operation.
entries	List<OrchestrationJobResponse>	List of orchestration job records.
count	Number	Total number of orchestration job records.

### 3.4 struct **OrchestrationJobResponse**

Field	Type	Description
id	OrchestrationJobId	Unique job identifier.
status	OrchestrationJobStatus	Actual working state of the job.
type	OrchestrationType	Type of orchestration.
requesterSystem	SystemName	Name of the system that started the orchestration process.
targetSystem	SystemName	Name of the system for which the orchestration is executed.
serviceDefinition	ServiceName	Name of the service that the orchestration job is targeting.
subscriptionId	OrchestrationSubscriptionId	Unique identifier of associated subscription record.
message	String	Additional error or warning information.
createdAt	DateTime	The job was created at this timestamp.
startedAt	DateTime	The job was started at this timestamp.
finishedAt	DateTime	The job was finished at this timestamp.

### 3.5 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.

### 3.6 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
DateTime	Pinpoints a specific moment in time.
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.
List<A>	An <i>array</i> 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.
OrchestrationJobId	Unique string identifier.
OrchestrationJobStatus	Predefined values indicating working states.
OrchestrationSubscriptionId	Unique string identifier.
OrchestrationType	Predefined values indicating orchestration type (pull or push).
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.



ARROWHEAD

Document title  
**serviceOrchestrationHistoryManagement**  
Date  
**2025-07-11**

Version  
**5.0.0**  
Status  
**DRAFT**  
Page  
**8 (9)**

## 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	