

orchestration-qos-reservations

Service Description

Abstract

This document provides service description for the **orchestration-qos-reservations** service.

Contents

| | |
|---|----------|
| 1 Overview | 3 |
| 1.1 How This Service Is Meant to Be Used | 3 |
| 1.2 Important Delimitations | 3 |
| 1.3 Access policy | 3 |
| 2 Service Interface | 4 |
| 2.1 interface HTTP/TLS/JSON | 4 |
| 3 Information Model | 5 |
| 3.1 struct QoSReservationList | 5 |
| 3.2 struct QoSReservation | 5 |
| 3.3 Primitives | 6 |
| 4 References | 7 |
| 5 Revision History | 8 |
| 5.1 Amendments | 8 |
| 5.2 Quality Assurance | 8 |

1 Overview

This document describes the **orchestration-qos-reservations** service which enables systems to get current provider reservations from the Orchestrator Core 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 Gatekeeper Core System should consume the Service Registry Core System's **query** service to get information about the **orchestration-qos-reservations** service. Using this information the system can get a list of provider reservations from the database of the Orchestrator Core System.

1.2 Important Delimitations

If the Quality-of-Service support is not enabled, this service always returns an empty list.

1.3 Access policy

This service is only available for the Gatekeeper Core System.

2 Service Interface

This section describes the interfaces to the service. The **orchestration-qos-reservations** service is used to get provider reservations. 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 () : QoSReservationList**

| 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 **orchestration-qos-reservations** 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 3.3, which are used to represent things like hashes and identifiers.

3.1 struct QoSReservationList

| Field | Type | Description |
|-------|----------------------|------------------------------------|
| count | Number | The number of reservation records. |
| data | List<QoSReservation> | Reservation records. |

3.2 struct QoSReservation

| Field | Type | Description |
|--------------------|------------|--|
| consumerAddress | Address | Network address of the consumer system. |
| consumerPort | PortNumber | Port of the consumer system. |
| consumerSystemName | Name | The name of the consumer system. |
| createdAt | DateTime | Reservation record was created at this UTC timestamp. |
| id | Number | Identifier of the record. |
| reservedProviderId | Number | The id of the provider system. |
| reservedServiceId | Number | The id of the service that is reserved. |
| reservedTo | DateTime | Expiration of the reservation. |
| temporaryLock | Boolean | Is this reservation temporary (with short expiration)? Temporary reservation should be extend or removed before the end of the orchestration process. |
| updatedAt | DateTime | Reservation record was modified at this UTC timestamp. |

3.3 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 <code>true</code> or <code>false</code> . |
| DateTime | Pinpoints a specific moment in time. |
| List<A> | An <i>array</i> of a known number of items, each having type A. |
| Name | A string identifier that is intended to be both human and machine-readable. |
| Number | Decimal number |
| PortNumber | A Number between 0 and 65535. |

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>



ARROWHEAD

Document title
orchestration-qos-reservations
Date
2023-02-27

Version
4.6.0
Status
RELEASE
Page
8 (8)

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 | |