# orchestration-create-flexible-store-rules

## Service Description

**Abstract**
This document provides service description for the **orchestration-create-flexible-store-rules** service.

# Contents

# 1 Overview

This document describes the **orchestration-create-flexible-store-rules** service which enables systems to add flexible matching rules to the Orchestrator Core System (if the Orchestrator is in flexible store mode).

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 Plant Description Engine Core System should consume the Service Registry Core System's **query** service to get information about the **orchestration-create-flexible-store-rules** service. Using this information the system can add a list of flexible store rules to the database of the Orchestrator Core System.

## 1.2 Important Delimitations

- Only works if the Orchestrator Core System is in flexible store mode. Otherwise, it returns with an error.
- Not support rules about providers from an other cloud.

## 1.3 Access policy

This service is only available for the Plant Description Engine Core System.

# 2 Service Interface

This section describes the interfaces to the service. The **orchestration-create-flexible-store-rules** service is used to add new rules to the Orchestrator's flexible store. 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 (List<FlexibleRule>) : FlexibleRuleListResponse

| 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-create-flexible-store-rules** 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 3.6, which are used to represent things like hashes and identifiers.

## 3.1 struct FlexibleRule

| Field | Type | Mandatory | Description |
|---|---|---|---|
| consumerSystem | SystemDescriptor | yes | Requirements for a system that tells if this rule is applied to a consumer. |
| priority | Number | no | Priority of the rule. |
| providerSystem | SystemDescriptor | yes | Requirements for a system that tells if this rule is applied to a provider. |
| serviceDefinition | Name | yes | Identifier of a service. |
| serviceInterfaceName | Interface | no | Interface requirement |
| serviceMetadata | Metadata | no | Service instance level metadata requirements. |

## 3.2 struct SystemDescriptor

| Field | Type | Mandatory | Description |
|---|---|---|---|
| systemName | Name | no (yes) | Name of a system. Mandatory if *metadata* is not specified. |
| metadata | Metadata | no (yes) | System level metadata requirements (a system's metadata must contain all of its keys with the same value to match). Mandatory if *systemName* is not specified. |

## 3.3 struct Metadata

An Object which maps String key-value pairs.

## 3.4 struct FlexibleRuleListResponse

| Field | Type | Description |
|---|---|---|
| count | Number | The number of created rules. |
| data | List<FlexibleRuleResponse> | Created rule records. |

## 3.5 struct FlexibleRuleResponse

| Field | Type | Description |
|---|---|---|
| consumerSystem | SystemDescriptor | Requirements for a system that tells if this rule is applied to a consumer. |
| createdAt | DateTime | Rule record was created at this UTC timestamp. |
| id | Number | Identifier of the rule. |
| priority | Number | Priority of the rule. |
| providerSystem | SystemDescriptor | Requirements for a system that tells if this rule is applied to a provider. |
| serviceDefinition | Name | Identifier of a service. |
| serviceInterface | Interface | Interface requirement |
| serviceMetadata | Metadata | Service instance level metadata requirements. |
| updatedAt | DateTime | Rule record was modified at this UTC timestamp. |

## 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. |
| Object | Set of primitives and possible further objects. |
| Interface | Any suitable type chosen by the implementor of service |
| List<A> | An *array* 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 |

# 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 | |