Service Discovery Register HTTP/TLS/JSON

Interface Design Description

Service ID: *"unregister"*

Abstract
This document describes a HTTP/TLS/JSON variant of the Service Discovery Unregister service.

ARTEMIS Innovation Pilot Project: Arrowhead
THEME [SP1-JTI-ARTEMIS-2012-AIPP4 SP1-JTI-ARTEMIS-2012-AIPP6]
[Production and Energy System Automation Intelligent-Built environment and urban infrastructure for sustainable and friendly cities]

Contents

1 Overview

This document describes the HTTP/TLS/JSON variant of the Service Discovery Unregister Eclipse Arrowhead service, which is enables autonomous service unregistration by systems. Examples of this interaction is a system that if going offline, or ceased to offer the capability to provide some kind of service. Since it was previously registered, to enable other systems to use, to consume it, this service needs to be unregistered from the Service Registry. A provider is allowed to unregister only its own services. It means that provider system name and certificate common name must match for successful unregistration.

This document exists as a complement to the *Service Discovery Unregister – Service Description* document. For further details about how this service is meant to be used, please consult that document. The rest of this document describes how to realize the Service Discovery Unregister service using HTTP [1], TLS [2] and JSON [3], both in terms of its functions (Section 2) and its information model (Section 3).

2 Service Functions

This section lists the functions that must be exposed by the Service Discovery Unregister service in alphabetical order. In particular, each subsection first names the HTTP method and path used to call the function, after which it names an abstract function from the Service Discovery Unregister SD document, as well as input and output types. All functions in this section respond with the HTTP status code `200 OK` if called successfully. The error codes are, `400 Bad Request` if request is malformed, `401 Unauthorized` if improper client side certificate is provided, `500 Internal Server Error` if Service Registry is unavailable.

2.1 DELETE /serviceregistry/unregister ?address={address} &port={port} &service_definition={service_definition} &system_name={system_name}

Interface:	Unregister
Input:	ServiceRegistryUnregisterRequest

Called to unregister a service offered by the caller system, as exemplified in Listing 1.

```
DELETE /serviceregistry/unregister?address=10.0.0.0&port=8080&service_definition=temperature&
    system_name=mytemperaturesensor HTTP/1.1
Accept: application/json
```

Listing 1: An Unregister invocation.

3 Information Model

Here, all data objects that can be part of the service calls associated with this service are listed in alphabetic order. Note that each subsection, which describes one type of object, begins with the *struct* keyword, which is meant to denote a JSON Object that must contain certain fields, or names, with values conforming to explicitly named types. As a complement to the primary types defined in this section, there is also a list of secondary types in Section 3.2, which are used to represent things like hashes, identifiers and texts.

3.1 struct ServiceRegistryUnregisterRequest

Identifies a requested Unregister call. As the fields of this type occur only as query parameters in the Unregister function, there is no need for it to representable as a JSON Object. This subsection exsits only for the sake of completeness.

3.2 Primitives

No primitives are required.

4 References

[1] R. Fielding and J. Reschke, “Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing,” RFC 7230, 2018, RFC Editor. [Online]. Available: https://doi.org/10.17487/RFC7230

[2] E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.3,” RFC 8446, 2018, RFC Editor. [Online]. Available: https://doi.org/10.17487/RFC8446

[3] T. Bray, “The JavaScript Object Notation (JSON) Data Interchange Format,” RFC 7159, 2014, RFC Editor. [Online]. Available: https://doi.org/10.17487/RFC7159

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	2020-12-05	1.0.0		Szvetlin Tanyi

5.2 Quality Assurance

No.	Date	Version	Approved by
1			