getWeatherSensors

Service Description

Abstract
This is the Service Description (SD document) for the ”getWeatherSensors” service according to the Eclipse Arrowhead documentation structure.

Contents

1 Overview

This document describes the ”getWeatherSensors” service, which send the weather measurements from the Weather Sensor System to the Controller.

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 ”getWeatherSensors” service is designed to provide weather-related measurements collected by the Weather Sensor System. This service facilitates seamless integration into larger systems, enabling users to retrieve environmental data for monitoring, analytics, or decision-making processes. Typical use cases include real-time weather monitoring for smart cities, predictive maintenance for infrastructure affected by weather conditions, and environmental research studies. The service operates in a synchronous request-response manner, ensuring that clients receive up-to-date sensor data whenever requested.

1.2 Important Delimitations and Dependencies

The ”getWeatherSensors” service has the following delimitations and dependencies:

- The service assumes that all sensors in the Weather Sensor System are functioning correctly and transmitting valid data. Faulty or offline sensors may result in incomplete or inaccurate responses.
- The data provided by the service is limited to the metrics defined in the WeatherSensorResponseDTO structure (temperature, humidity, pressure, and wind speed).
- This service does not perform advanced data processing or analysis; it is focused solely on retrieving and transmitting raw sensor data.

Furthermore, the service does not handle:

- Error recovery or retry mechanisms in case of network failures; such functionality must be implemented at the system level.
- Initial setup or configuration of the lamps in the system. It assumes all lamps are already registered in the database.
- Physical hardware issues, such as faulty lamps or communication modules, which must be managed by maintenance protocols external to this service.

2 Service Interface

This section describes the interfaces to the ”getWeatherSensors” service. In particular, each subsection names an abstract operation, 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 interface operations are available.

2.1 operation getWeatherSensors () : List <WeatherSensorResponseDTO>

The getWeatherSensors operation is used to send the data from the sensors as a list of WeatherSensorResponseDTO.

3 Information Model

Here, all data objects that can be part something the ”getWeatherSensors” Service provides to the hosting System are listed in alphabetic order.

3.1 struct WeatherSensorResponseDTO

This structure is used to store all the sensors data.

Field	Type	Description
id	String	Sensor ID.
temperature	String	Temperature measured (℃).
humidity	String	Humidity measured (percentage).
pressure	String	Pressure measured (hPa).
wind	String	Wind speed measured (km/h.

3.2 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 `true` or `false`.
Interface	Any suitable type chosen by the implementor of the service.
DateTime	Pinpoints a specific moment in time.
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.
PortNumber	Decimal number in the range of 0-65535
Version	Specifies a service version.

4 References

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	2020-12-05	X.Y.Z		Tanyi Szvetlin
2	2021-07-14	X.Y.Z	Minor updates	Jerker Delsing
3	2022-01-10	X.Y.Z	Minor updates	Jerker Delsing

5.2 Quality Assurance

No.	Date	Version	Approved by
1	2022-01-10	X.Y.Z	