dynamicPrice

Service Description

Abstract
This Service Description (SD) defines the dynamicPrice service, produced by the DynamicPricingSystem.

Contents

1 Overview

The dynamicPrice service exposes the current parking price computed by the DynamicPricingSystem. The service allows client systems to obtain updated price information for display or control purposes. It uses contextual inputs such as occupancy, time, weather and local events to determine the optimal rate.

This service is typically consumed by the ClientAppSystem and other subsystems that require current price data.

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 service is intended for use inside the Arrowhead local cloud. The client app make a request to the dynamicPrice endpoint, optionally including metadata such as parking zone (electrical or regular parking) or timestamp. The service returns the current price and its validity window.

1.2 Important Delimitations and Dependencies

The service does not handle payment, bookings or authentication. It depends on valid inputs from OccupancySystem, WeatherSystem, EventSystem and TimeSystem for accurate price calculation.

2 Service Interface

This section describes the interfaces to the dynamicPrice 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 getDynamicPrice (priceRequest) : priceResponse

The getDynamicPrice operation returns the most recent price for a parking area. The request may include optional metadata such as parking zone ID and timestamp. The response includes the current price and validity period.

3 Information Model

Here, all data objects that can be part something the XX 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 priceRequest

This structure defines the input parameters for a price request.

Field	Type	Description
parkingZoneID	Name	Identifier for the parking area
timestamp	DateTime	Time of the request

3.2 struct priceResponse

This structure defines the service output that contains the dynamic price and related information.

Field	Type	Description
priceValue	Float	Calculated price in local currency
validFrom	DateTime	Start time of the validity window
validUntil	DateTime	End time of the validity window

3.3 Primitives

Type	Description
Name	A string identifier
DateTime	A timestamp expressed in UTC format
Float	A decimal number representing a continuous value

4 References

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	2025-10-14	1.0.0	Initial release	Mattias Öhman