dynamicPrice HTTP

Interface Design Description

Abstract
This Interface Design Description (IDD) defines the HTTP implementation of the dynamicPrice service. The interface uses secure HTTPS with JSON payloads to provide dynamic parking prices within the local cloud.

Contents

1 Overview

This document describes the HTTP/JSON interface for the dynamicPrice service produced by the DynamicPricingSystem. The interface allows the client app system to request the current parking price. Communication occurs over HTTPS with JSON payloads.

Profile type	Type	Version
Transfer protocol	HTTPS	1.1
Data encryption	TLS	1.3
Encoding	JSON	RFC 8259 [**?**]
Compression	N/A	-
Semantics	SensML	RFC 9100
Ontology	N/A	-

Table 1: Communication profile for the dynamicPrice service interface

This document provides the Interface Design Description IDD to the *dynamicPrice – 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 dynamicPrice service interface in detail. Both in terms of its operations (Section 2) and its information model (Section 3).

2 Service Operations

2.1 POST /dynamicPrice

Operation: getDynamicPrice
Input: priceRequest
Output: priceResponse

Called by the ClientAppSystem to obtain the current parking price for a given parking zone. The request and response are encoded in JSON format. The POST method is used to allow structured input containing optional fields such as timestamp and zone ID, following Arrowhead conventions.

```
POST /dynamicPrice HTTP/1.1

{
  "parkingZoneID": "zone_A1",
  "timestamp": "2025-10-14T08:30:00Z"
}
```

Listing 1: Example request to getDynamicPrice

```

{
  "priceValue": 12.50,
  "validFrom": "2025-10-14T08:30:00Z",
  "validUntil": "2025-10-14T09:30:00Z"
}
```

Listing 2: Example response from getDynamicPrice

3 Data Models

3.1 struct priceRequest

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

3.2 struct priceResponse

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	String identifier for parking zone
DateTime	Timestamp in ISO 8601 format (UTC)
Float	Decimal value representing a numeric amount

3.3.1 alias Name = String

A string identifier that uniquely names a parking area or zone. Example: ”zone_A1”.

3.3.2 alias DateTime = String

Pinpoints a moment in time in the format of ”YYYY-MM-DD HH:mm:ss”, where ”YYYY” denotes year (4 digits), ”MM” denotes month starting from 01, ”DD” denotes day starting from 01, ”HH” denotes hour in the 24-hour format (00-23), ”MM” denotes minute (00-59), ”SS” denotes second (00-59). ” ” is used as separator between the date and the time. An example of a valid date/time string is ”2020-12-05 12:00:00”

3.3.3 alias Float = Number

Represents a decimal number using a period as separator, for example: 12.50.

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