Guidance for Payment and Permit API development

ASURA Technologies Ltd.
asuratechnologies.com
H-1122, Budapest
Városmajor utca 12-14. C épület

ASURA

Table of Contents

1. Introduction

This document details the API standards necessary for effective integration and data management between ASURA Compliance system and the Payment Data providers, focusing on location data, payment transactions, and permit information. Adhering to these standards ensures seamless integration, consistent data handling, and efficient communication between our systems and those of our partners. By following these guidelines, we aim to achieve reliable and accurate data exchange, which is crucial for maintaining operational integrity and delivering excellent service to our users. The following sections provide the specifications, requirements, and examples needed to meet these API expectations.

1.1. General requirements

The data collected through these APIs is crucial for compliance and enforcement purposes, and the information required can be divided into two main parts.

The first API is designed to deliver **location information**, which forms the basis for further data handling.

Depending on the type of payment or permit involved, we expect to implement either a transactional API or a permit API. These APIs will address different aspects of data exchange:

- the transactional API will manage payment-related transactions,
- while the permit API will handle permit-related information.

Although both APIs will share some similarities, they also have distinct functions, which will be detailed later in the document.

The general approach is to first provide information on active locations where payment transactions occur. Once these locations are identified, both the payment and permit APIs can be utilized to pull relevant data for these locations, ensuring a comprehensive and accurate flow of information.

1.2. API communication method

To ensure compatibility and efficient integration, the API should adhere to the following specifications:

- **Preferred Method**: HTTP API
- **Endpoints**: The API should include 2 or 3 endpoints for the following services: location, payment, and permit.
  - GET Location
  - GET Transaction
  - GET Permit
- **HTTP Method**: GET
- **Data Format**: JSON
- **Authentication and Authorization**: The API must support authentication and authorization via API keys, OAuth, or Basic Authentication.
- **Parameters for Transactions and Permits APIs**

- start_time: UNIX timestamp //seconds
- end_time: UNIX timestamp //seconds
- locaton: location ID //refers to Location API – location, optional

Transaction Retrieval Requirement: Based on the given parameters, **i**t should retrieve all transactions that were active during the specified time period, including those that began before the period but were still active within it, as well as transactions that started during the period.

2. Location API

The Location API is designed to provide a complete list of locations accessible with the given authentication credentials. It returns each location's unique Location ID, which can be used as a parameter for accessing other APIs. Additionally, the API includes relevant business data associated with each location. This data enables users to retrieve and utilize location-specific information efficiently.

Example

HTTP GET

- Header:
- x-api-key : GUID //authentication key

URL:

- https://sample.provider.com/services/Location

Response:

```
"locations": [
        {
                "lot_number": "3875", //required, unique
                "lot_name": "Avenue E & Third Gas Station", //required
                "lot_address": "300 Avenue E",
                "lot_city": "San Antonio",
                "lot_state": "TX",
                "lot_zipcode": "78205",
                "lot_timezone": "America/Chicago"
        },
        {
                "lot_number": "3876",
                "lot_name": "Avenue C ",
                "lot_address": "301 Avenue C",
                "lot_city": "San Antonio",
                "lot_state": "TX",
                "lot_zipcode": "78206",
                "lot_timezone": "America/Chicago"
```

```
        }...
]
```

3. Transaction API

The Transaction API provides detailed information on transactions and reservations for a specified location ID within a given period. It retrieves data on all hourly and daily transactions, including their current status, vehicle details, ticket prices, and buyer information.

The API also supports the handling of various changes related to transactions:

- **Status Changes:** Allows updates such as cancellation of a ticket before its validity date.
- **Plate Changes:** Enables modifications to the vehicle plate number before the ticket is used.
- **Date and Time Changes:**
  - Changes to the purchase date before the ticket's validity period begins.
  - Adjustments to the expiration date before the ticket's active time ends.
- **Other Modifications:** Transaction adjustments that do not affect the ticket's actual usage at any time.

This API ensures that all transaction-related data is up-to-date and accurately reflects the current state of reservations and ticket usage.

3.1. Required fields

To ensure the Compliance system receives all essential data, the API response must include the following minimum information (mandatory)

1. **Transaction ID**: A unique identifier for the transaction.
2. **Location ID**: Corresponds to the lot_number from the Location API.
3. **Transaction Status**: Indicates the current status (Active or Cancelled).
4. **Plate Number**: The vehicle's license plate number.
5. **Purchase Date**: The start date and time (UTC) of the transaction.
6. **Expiration Date**: The end date and time (UTC) of the transaction.
7. **Rate**: The value or price of the transaction.
8. **Created Date**: The date (UTC) when the transaction was created.
9. **Modification Date**: The date (UTC) of any modifications made to the transaction.

If the provider supports active plate changes, where customers can update the plate number during the active period of the transaction, this should be documented as follows:

10. **Plate History**: Array that contains records of plate changes, including:
    Plate Number: The new plate number.
    Created Date: The date when the plate change became effective.

3.2. Optional fields

Additional business data to include (optional):

1. **Plate Country**: The country of registration for the plate.
2. **Plate State**: The state or province of registration for the plate.
3. **Owner Name**: The name of the vehicle owner.
4. **E-mail**: The owner's email address.
5. **Phone Number**: The owner's contact phone number.
6. **Vehicle Make**: The manufacturer of the vehicle.
7. **Vehicle Model**: The model of the vehicle.
8. **Vehicle Color**: The color of the vehicle.

This comprehensive data will ensure accurate tracking and management of transactions within the Compliance system.

4. Permit API

The Permit API offers comprehensive details about long-term permits for a specified location ID within a given timeframe. It provides information on all permits, including their current status, vehicle details, ticket prices, strictions, and buyer information.

The Permit API can accommodate three approaches:

1. **Single Location, Single Permission:**
   - Use the Location API and Permit API.
   - Location ID and Timezone can be omitted from the Permit API, as these are covered by the Location API.
2. **Multiple Locations, Single Permission Each -** When a permit is valid at different locations but has the same permissions at each location:
   - Include the Location ID and Timezone in each permission entry within the Permit API.
   - This approach bypasses the need to reference the Location API for location-specific details.
3. **Multiple Locations, Multiple Permissions -** When a permit is valid at various locations, each with potentially different permissions:
   - Include the Location ID and Timezone for each permission entry within the Permit API.
   - This ensures that each permission's specific location details are clearly defined without relying on the Location API.

By following these approaches, the Permit API can flexibly handle different scenarios, ensuring that location and permission details are accurately represented and easily accessible.

Validations performed

- The external ID must be populated (i.e. it cannot be empty, null, or contain only whitespace).
- The start date is mandatory and must be in ISO-formatted UTC.

- Each permit must contain at least one vehicle.
- Each permit must contain at least one time restriction.
- The vehicle's start and end date interval must be a subset of the permit's start and end date interval.
- The vehicle start date is optional, but if an end date is provided, then a start date must also be specified.
- The relation vehicle start date ≤ vehicle end date must hold true.
- For time restrictions, the following is expected:
  - The valid from and valid to days must be specified.
  - The time values must be valid (e.g. 11:12 is valid, but 25:00 is not).
  - The location to which each time restriction applies must be indicated.

Key Features:

- **Time Restrictions:** Specifies when the permit can be used on a given day. Restrictions can be defined in two ways:
- **Daily Validity:** Permits valid for any time within a 24-hour period on specific days.
- **Business Hours:** Permits valid only during defined business hours.
- **Multi-Plate Support:** Manages permits associated with multiple plate numbers. While multiple plates may be linked to a single permit, only one vehicle can use the facility at a time. Any overlapping use by multiple vehicles would be considered non-compliant.

Supported Changes:

- **Status Changes:** Allows updates such as cancellations before the permit's validity begins.
- **Plate Changes:** Facilitates modifications to vehicle plate numbers before or during the permit's usage.
- **Date and Time Changes:**
  - **Purchase Date:** Adjustments can be made before the permit's validity period starts.
  - **Expiration Date:** Modifications can be applied before the permit's active time ends.
  - **Time Restrictions:** Updates can be made to time restrictions, but only for future dates.
- **Other Modifications:** Permits changes that do not impact the actual usage of the permit at any time.

This API ensures that permit-related data is accurately maintained, reflecting the current status and details of permits, and supports effective management of permits and their associated conditions.

4.1. Required fields

To ensure the Complaince system receives all essential data, the API response must include the following mandatory information:

1. **Permit ID:** A unique identifier for the permit.
2. **Vehicles:** An array containing records of:

Plate Number: The vehicle's license plate number.
Plate Country: The Country of the vehicle's license plate.
Plate State: The state of the vehicle's license plate
Vehicle Make: The manufacturer of the vehicle.
Vehicle Model: The model of the vehicle.
Vehicle Color: The color of the vehicle.
Start Date: The date (UTC) when the vehicle is added to the permit.
End Date: The date (UTC) when the vehicle is removed from the permit.

3. **Purchase Date:** The start date and time (UTC) of the permit.
4. **Expiration Date:** The end date and time (UTC) of the permit.
5. **Permissions with Time Restrictions:** An array describing the days and times when each permission is valid for parking at each location. Each location can have several permissions with different time restrictions:
   Permission Name: Describes the name of the permission.
   Permission Type: Describes the type of the permission, if applicable.
   Days: The days when the permission is valid. If empty, it applies to all days. E.g. Days: ["Monday", "Sunday"]
   Time Begin: The start time of entry in the parking lot's local time (HH:MM format). If empty, it is an all day permit.
   Time End: The end time of exit in the parking lot's local time (HH:MM format). If empty, it is an all day permit.
   Location ID: Corresponds to the `lot_number` from the Location API.
   TimeZone: The timezone of the location.
6. **Rate:** The value or price of the permit.
7. **Created Date:** The date (UTC) when the permit was created.
8. **Modification Date:** The date (UTC) of any modifications made to the permit.

4.2. Optional fields

Additional business data to include (optional):

1. **Account Number**: Account number of the permit holder
2. **Permit Type**: The type of the permit
3. **Permit Type ID**: The ID of the permit type
4. **Plate Country**: The country of registration for the plate.
5. **Plate State**: The state or province of registration for the plate.
6. **Owner Name**: The name of the vehicle owner.
7. **E-mail**: The owner's email address.
8. **Phone Number**: The owner's contact phone number.

This comprehensive data will ensure accurate tracking and management of transactions within the Compliance system.

5. Technical documentation

Sent as a separate attachment.