

- [Documentation](#) 
- [1. Introduction](#) 
- [2. Categories](#) 
- [3. V2.0 Interface Documentation \(HTTP\)](#) 
 - [3.1 Development Guide](#) 
- [4. Query Interfaces \(V2.0\)](#) 
 - [Query Zone Information](#) 
 - [Query Booth Information](#) 
 - [Query Lane Information](#) 
 - [Query Plate Type Information](#) 
 - [Query Plate Color Information](#) 
 - [Query Lane Ground Loop Barrier Status](#) 
 - [Query Lane Current Order](#) 
 - [Query Vehicle Record](#) 
 - [Query Blacklist Records](#) 
 - [Query Parking Order Records](#) 
 - [Query Payment Records](#) 
 - [Query Payment Detail Records](#) 
 - [Query Gate Release Record](#) 
 - [Query Coupon Scheme](#) 
 - [Query Plate Coupon Records](#) 
 - [Query Parking Fee for Specified Order](#) 
- [5. Notification Interfaces \(V2.0\)](#) 
 - [In-Parking Record Change Notification \(Including Modify and Close\)](#) 
 - [Notify Vehicle Changes \(Including Add, Edit, Delete\)](#) 
 - [Notify Blacklist Changes \(Including Add, Edit, Delete\)](#) 
 - [Notify Plate Coupon Record Changes \(Including Add, Edit, Delete\)](#) 
 - [Notify Coupon Scheme Changes \(Including Add, Edit, Delete\)](#) 
 - [Notify Control Lane Barrier Status \(Including Release, Close, Normally Release, Cancel Normally Release\)](#) 
 - [Notify Lane Popup Order Fee Modification](#) 
 - [Notify Vehicle Payment Information](#) 
 - [Create Entry Order for Vehicle Without Plate](#) 
 - [Notify Callback Parameter Changes](#) 
 - [Notify Failed Callback Business Retry](#) 
- [6. Callback Interfaces \(V2.0&V3.0\)](#) 
 - [Vehicle Entry Callback](#) 
 - [Vehicle Exit Callback](#) 
 - [Payment Result Callback](#) 
 - [Vehicle Issuance Callback](#) 
 - [Vehicle Cancellation Callback](#) 
 - [Vehicle Recharge and Extension Callback](#) 
 - [Gate Release Record Callback](#) 

- [Parking Space Change Callback](#) 
- [Blacklist Callback](#) 
- [Booth Lane Information Callback](#) 
- [7. V3.0 Interface Documentation \(MQTT\)](#) 
 - [7.1 Development Guide](#) 
 - [7.2 Interface Route & Difference Table](#) 

Parking Software LAN Interface

Documentation

This document includes an introduction to the LAN interface, classification and prerequisites for using the LAN interface, and explanations of the corresponding interface documentation for the LAN interface.

1. Introduction

The LAN interface is designed for third-party integration scenarios where parking lot-related data is not intended to be uploaded to the platform, but there is a need to perform related data queries, additions, modifications, and receive entry/exit data from the software side within the local area network.

The LAN interface is divided into three categories: **Query**, **Notification**, and **Callback**.

- **Query** : Refers to requesting relevant data from the software side via the interface, such as querying records of vehicles in the lot or payment records.
- **Notification**: Refers to sending relevant data to the software via the interface, after which the software saves the data, such as adding a license plate or controlling the gate to open.
- **Callback**: Refers to the software uploading relevant data to the integrator's server, where the integrator is responsible for saving this type of data, such as uploading available parking spaces, entry information, or exit information.

2. Categories

The LAN interface is divided into two versions: **V2.0** and **V3.0**:

- **V2.0 interface**: This version offers a more comprehensive set of functional interfaces and provides greater convenience for interface invocation.
- **V3.0 interface**: This version utilizes the MQTT protocol for integration, effectively overcoming network access restrictions and ensuring real-time and reliable data interaction.

3. V2.0 Interface Documentation (HTTP)

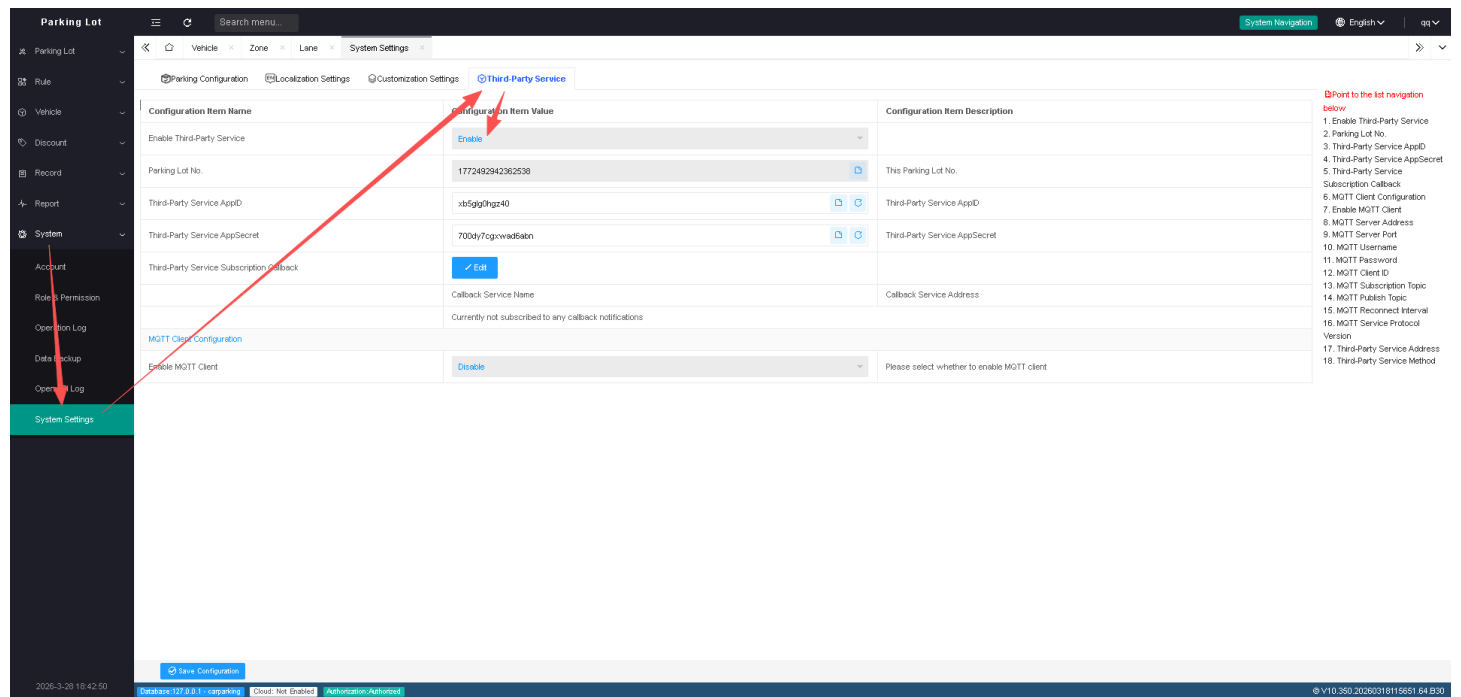
This service is the LAN interface designed for third-party [developers](#)  calling the API interfaces we provide, developers can integrate the following functions into their own applications or services: querying information related to local parking lot operations, parking payment, vehicle registration, Gate Release Record, and other features.

3.1 Development Guide

This document primarily covers the acquisition of parameters and software-side configurations during integration, as well as basic rules and related type enumerations (payment methods, license plate colors, callback interfaces, payment scenarios, order statuses, query operators, etc.). It also includes interface signature methods, handling of sensitive information, callback configuration instructions, and more.

3.1.1 Parameter Acquisition and Configuration

Calling the API requires the use of the appid, AppSecret and ParkKey parameters. These must be configured and filled in within the System Settings. The specific steps are shown below:



3.1.2 Basic Rules

Basic Information

All API requests must use `HTTP`.

Character Set

Interfaces use the `UTF-8` character encoding.

Field Naming Rules

All parameter fields related to the interface adopt Pascal Case naming convention. Similar to Camel Case but with the first letter capitalized, e.g., `FirstName`, `LastName`, `UserName`.

Request Unique Identifier

The parking lot's local open interface assigns a unique identifier to each received request. This unique identifier is included in the response's `ResultId`. When assistance with parking lot interface development is needed, please provide the request's unique identifier and relevant parking lot logs to help us locate the specific request more quickly.

Date Format

All date objects follow the format examples defined below:

```
1 | yyyy-MM-dd HH:mm:ss
2 | yyyy-MM-dd HH:mm:ss:fff
3 | yyyy-MM-dd
```

Parameter Format

Interface request parameter type uses `Content-Type=application/json` .

3.1.3 Interface Authentication

Overview

- This document provides a detailed description of the signature-logic implementation for open-interface requests to the parking-lot system.
- The logic covers request-parameter processing, sorting, and HMAC-SHA256 signing.

Signature Generation Steps

1. Prepare data

- Collect the items that will take part in the signature: `AppId` , `ParkKey` , `TimeStamp` , `Nonce` , `Data` .

2. Process `Data`

- Sort every key/value pair in `Data` by ASCII code (ascending).
- Concatenate the pairs with "&" and convert the whole string to **lower-case**.
- Compute the **MD5** hash of the resulting string (32-char hex, UTF-8).

Rules

- If `Data` is an array or collection, convert it to a list, process each element individually, then join the results with "&".
- If `Data` is a single object, sort its fields directly.
- When `Data` is `null` , still include it in the signature as `Data=` (empty value).
Any other parameter whose value is `null` is **omitted** from the signature string.

3. Build the signature string

- Concatenate the following five items in **exact order**, separated by "&": `AppId` , `Data` (already hashed), `ParkKey` , `TimeStamp` , `Nonce` .

4. Sign

- Convert the entire concatenated string to **lower-case**.
- Compute **HMAC-SHA256** using `AppSecret` as the key.
- Encode the digest as **Base64** and convert it to **lower-case**.
- The result is the final signature value.

Example

Given the request body:

```
1  {
2    "ParkKey": "1702399313365243",
3    "AppId": "fkqju0n1",
4    "TimeStamp": "1704527859009",
5    "Nonce": "928885",
6    "Sign": "x1aeqmo3/hs9cbhlbnzburgskf1gvd/lil2t/iasiy=",
7    "Data": [
8      { "Index": 1, "Url": "http://aebhfqld.gt/tgmjt " },
9      { "Index": 3, "Url": "http://qtdupkzq.sd/dsicxsc " },
10     { "Url": "http://dfxkmte.tv/mpeop ", "Index": 2 },
11     { "Url": "http://hdphhisif.ni/uejufkem ", "Index": 4 }
12   ]
13 }
```

Step 1 – normalise Data

Sort all key/value pairs by key (ASCII ascending):

```
Index=1&Url=http://aebhfqld.gt/tgmjt&Index=2&Url=http://dfxkmte.tv/mpeop&Index=3&Url=http://qtdupkzq.sd/dsicxsc&Index=4&Url=http://hdphhisif.ni/uejufkem
```

Lower-case the string and compute MD5 → assume the digest is

```
md5Result = 5d41402abc4b2a76b9719d911017c592
```

Step 2 – build the pre-sign string

```
AppId=fkqju0n1&Data=5d41402abc4b2a76b9719d911017c592&ParkKey=1702399313365243&TimeStamp=1704527859009&Nonce=928885
```

Step 3 – sign

- Lower-case the entire string and run HMAC-SHA256 with `AppSecret` .
- Base64-encode the digest and lower-case it → the value must match the `Sign` field in the request.

Remarks

- Every key/value pair must be sorted by ASCII code.
- The HMAC key is always `AppSecret` .
- `TimeStamp` must be a **13-digit Unix timestamp** (milliseconds).
- `Nonce` must be a random string to prevent replay attacks.
- When `Data` is `null` , still include `Data=` in the signature string; all other `null` fields are skipped.
- In callback interfaces, some image Base64 fields are not included in signature calculation. The specific field remarks shall prevail.

Code Samples

HMAC-SHA256 helper (C#)

```
1    /// <summary>
2    /// HMAC-SHA256 signature
3    /// </summary>
4    /// <param name="data">data to be signed</param>
5    /// <param name="secret">secret key</param>
6    /// <returns>signed data, base64 encoded, converted to lower-case</returns>
7    public static string HmacSha256Sign(string data, string secret)
8    {
9        try
10        {
11            var encoding = new UTF8Encoding();
12            var keyByte = encoding.GetBytes(secret);
13            var messageBytes = encoding.GetBytes(data);
14            using var hmacsha256 = new System.Security.Cryptography.HMACSHA256(keyByte);
15            var hashmessage = hmacsha256.ComputeHash(messageBytes);
16            var result = Convert.ToBase64String(hashmessage);
17            return result.ToLower();
18        }
19        catch (Exception ex)
20        {
21            OpenApiLog.WriteLog("HMAC-SHA256 signature exception", ex);
22            return string.Empty;
23        }
24    }
```

#####Convert Data to sorted string (C#)

```
1    /// <summary>
2    /// Sort Data by ASCII code (ascending) and convert to string
3    /// </summary>
4    /// <param name="data">request data</param>
5    /// <returns>converted string</returns>
6    private static string ConvertDataToString(object data)
7    {
8        // if data is null, return empty string
9        if (data == null)
10        {
11            return string.Empty;
12        }
13
14        // if data is array or collection
15        if (data.GetType().IsArray || data is IEnumerable<object>)
16        {
17            // convert array to list
18            var dataList = ((IEnumerable<object>)data).ToList();
19            // process each element in the list recursively
20            var dataDictList = dataList.Select(ConvertDataToString);
21            // use stable sort to ensure identical elements do not swap positions
22            dataDictList = dataDictList.OrderBy(p => p, StringComparer.Ordinal).ToList();
23            // convert each element to string and join with "&"
24            return string.Join("&", dataDictList);
25        }
26
27        // if data is object
28        // use reflection to get property values and convert to key-value pair sequence
29        var dataDict = data.GetType().GetProperties()
```

```

30         .Select(p =>
31         {
32             // get property value
33             var value = p.GetValue(data);
34             // if property value is array, process recursively
35             if (value != null && (value.GetType().IsArray || value is
IEnumerable<object>))
36             {
37                 value = ConvertDataToString(value);
38             }
39             // return key-value pair
40             return new KeyValuePair<string, object>(p.Name, value);
41         })
42         // filter out key-value pairs whose value is null
43         .Where(p => p.Value != null)
44         // use stable sort on key-value pair sequence
45         .OrderBy(p => p.Key, StringComparer.Ordinal)
46         // convert key-value pair sequence to dictionary
47         .ToDictionary(p => p.Key, p => p.Value);
48
49         // convert dictionary to string and join with "&"
50         return string.Join("&", dataDict.Select(p => $"{p.Key}={p.Value}"));
51     }

```

3.1.4 Type Enumeration List

LicensePlateColorEnum

No.	License Plate Color
0	Unknown
1	Blue
2	Yellow
3	White
4	Black
5	Green
6	Yellow-Green

PaymentMethodEnum

Payment Method Code	Payment Method Name
79001	Offline Cash Payment
79002	Platform Cash Payment
79003	WeChat Pay
79004	Android App Payment
79006	Terminal Device Payment

Payment Method Code	Payment Method Name
79007	Alipay
79009	Third-Party Payment
79010	Offline WeChat Pay
79011	Offline Alipay
79014	CCB Payment
79015	CMB One-Net Payment
79016	UnionPay Express Payment
79017	CCB Express Payment
79012	Alipay Express Payment
79013	WeChat Express Payment
79018	WeFuTong Aggregate Pay
79019	CMB Express Payment
79020	ICBC Express Payment
79021	ICBC Payment
79022	ABC Express Payment
79023	ABC Payment
79024	ETC Payment
79025	BOC Payment
79026	BOC Express Payment
79027	LeJuHe Payment
79028	UnionPay Business
80002	Self-Service Cash Payment
79029	Charging Deduction
79039	Suixingfu

CallbackInterfaceEnum

No.	Callback Interface Description	Supports Failure Retry
0	Vehicle Entry Callback	Yes
1	Vehicle Exit Callback	Yes
2	Payment Result Callback	Yes

No.	Callback Interface Description	Supports Failure Retry
3	Monthly Card Issuance Callback	Yes
4	Monthly Card Recharge/Extend Callback	Yes
5	Monthly Card Cancellation Callback	Yes
6	Parking Space Change Callback	No (latest data only)
7	Abnormal Exit Record Callback	Yes
8	Blacklist Info Callback	Yes
9	Violation Reservation Callback	Yes
10	Booth Lane Info Callback	No (query manually)

RequestResultCodeEnum

Result Code	Description
200	Operation successful
1001	Signature verification failed
1002	Invalid request parameters
1003	Interface service processing exception, please contact administrator
1004	Interface service not enabled
1005	Expected error, refer to actual error message returned

PaymentScenarioEnum

Payment Scenario Code	Payment Scenario Description
0	Other
1	In-Park QR Payment
2	Lane QR Payment

QueryOperatorEnum

Enum	Value	Description
Equal	0	Equal to
Like	1	Fuzzy query
GreaterThan	2	Greater than
GreaterThanOrEqual	3	Greater than or equal to
LessThan	4	Less than

Enum	Value	Description
LessThanOrEqual	5	Less than or equal to
In	6	In operation Correct format: x, y, z Wrong format: 'x', 'y', 'z'
NotIn	7	Not in operation; parameters same as In
LikeLeft	8	Left fuzzy
LikeRight	9	Right fuzzy
NoEqual	10	Not equal
IsNullOrEmpty	11	Is null or ''
IsNot	12	Case 1: Value $\neq$ null $\rightarrow$ field = x Case 2: Value = null $\rightarrow$ field is null
NoLike	13	Negation of fuzzy query
EqualNull	14	Case 1: Value $\neq$ null $\rightarrow$ field = x Case 2: Value = null $\rightarrow$ field is null
InLike	15	Correct format: x, y, z Wrong format: 'x', 'y', 'z' Generated SQL: id like '%X%' or id like '%Y%' or id like '%Z%'

Field Notice

Fields inside arrays returned by query-data interfaces **do not** support conditional filtering!

Operator Notes

MySQL field types and the operators they support:

1. Numeric Types

- Integer (INT , BIGINT , TINYINT , etc.)
Operators: = , <> , > , < , >= , <= , IN , NOT IN
- Floating-point (FLOAT , DOUBLE)
Operators: = , <> , > , < , >= , <= , IN , NOT IN

2. Character Types

- CHAR , VARCHAR
Operators: = , <> , LIKE , NOT LIKE , IN , NOT IN
- TEXT
Operators: = , <> , LIKE , NOT LIKE , IN , NOT IN

3. Date & Time Types

- DATE
Operators: = , <> , > , < , >= , <= , IN , NOT IN
- TIME
Operators: = , <> , > , < , >= , <= , IN , NOT IN
- DATETIME , TIMESTAMP
Operators: = , <> , > , < , >= , <= , IN , NOT IN

4. Boolean Type

- BOOLEAN
Operators: = , <> , IN , NOT IN

OrderStatusEnum

Order Status Code	Order Status Description
199	Pre-Entry
200	Entered
201	Exited
202	Auto Closed

ManualGateOperationTypeEnum

Manual Gate Operation Type Code	Manual Gate Operation Type Description
0	Exit opened without entry record
1	Manual open
2	Special plate number
3	Manual close
4	Barrier kept open / cancel keep-open
6	Illegal open

AbnormalPassageTypeEnum

Abnormal Passage Type Code	Abnormal Passage Type Description
6100	Auto release without entry record
6201	Manual open
6202	Manual close
6211	Barrier kept open

Abnormal Passage Type Code	Abnormal Passage Type Description
6212	Cancel keep-open

3.1.5 Sensitive Information Decryption

Description

When sensitive information is returned by a query, it is encrypted and must be decrypted.

How to Tell if a Field Needs Decryption

Encrypted fields are usually returned in `Base64` format and are marked “requires decryption” in the field description.

Decryption Method

Data is encrypted/decrypted with `AES`. Key details are in the table below:

Item	Description
Key (UTF-8)	infoexchange4api
IV (UTF-8)	activevector4api
Encoding	Base64
Key size	128-bit
Mode	CBC
Padding	Pkcs7 / Pkcs5

Sample Code

C# example

```
1      /// <summary>
2      /// Decrypts a value using the AES algorithm.
3      /// </summary>
4      /// <param name="value">Encrypted value to decrypt.</param>
5      /// <param name="key">Encryption key.</param>
6      /// <param name="iv">Initialization vector.</param>
7      /// <returns>Decrypted value as a string.</returns>
8      public static string AesDecrypt(string value, string key, string iv)
9      {
10         // Return input as-is if null or empty
11         if (string.IsNullOrEmpty(value)) return value;
12
13         // Validate key
14         if (key == null) throw new Exception("Encryption key not set.");
15         if (key.Length < 16) throw new Exception("Key length must be at least 16
16 characters.");
17         if (key.Length > 32) throw new Exception("Key length must not exceed 32
18 characters.");
19         if (key.Length != 16 && key.Length != 24 && key.Length != 32)
20             throw new Exception("Key length is ambiguous.");
21
22         // Validate IV length if provided
23         if (!string.IsNullOrEmpty(iv) && iv.Length < 16)
```

```

22         throw new Exception("IV length must be at least 16 characters.");
23
24         // Convert key to byte array
25         var keyBytes = Encoding.UTF8.GetBytes(key);
26
27         // Convert Base64-encoded cipher text to byte array
28         var encryptedBytes = Convert.FromBase64String(value);
29
30         // Create RijndaelManaged instance
31         using (var aes = new RijndaelManaged())
32         {
33             // Set IV: use provided value or first 16 bytes of key
34             aes.IV = !string.IsNullOrEmpty(iv) ? Encoding.UTF8.GetBytes(iv) :
keyBytes.Take(16).ToArray();
35
36             // Set encryption key
37             aes.Key = keyBytes;
38
39             // Set cipher mode to CBC
40             aes.Mode = CipherMode.CBC;
41
42             // Set padding mode to PKCS7
43             aes.Padding = PaddingMode.PKCS7;
44
45             // Create decryptor
46             var cryptoTransform = aes.CreateDecryptor();
47
48             // Transform cipher text back to original bytes
49             var resultArray = cryptoTransform.TransformFinalBlock(encryptedBytes, 0,
encryptedBytes.Length);
50
51             // Convert result bytes to UTF-8 string
52             return Encoding.UTF8.GetString(resultArray);
53         }
54     }

```

3.1.6 Callback Configuration Instructions

1. When is a callback service required?

When integrating third-party parking services, if you need the parking system to notify you of vehicle entry/exit, payment details, monthly ticket registration, monthly ticket extensions, monthly ticket cancellations, or parking space updates by area, you must configure a callback service.

2. How to configure the callback service

During integration, the developer must call the local parking service's "Set Callback Parameters" API to enable the required callback interfaces. If no parameters are set, no callbacks will be triggered by default.

3. What functions must the callback service implement?

The callback service must expose the APIs described in the callback interface documentation and return data in the specified format. To ensure data integrity, signature verification is recommended.

4. Callback service limitations

If the parking system fails to deliver a callback notification, it will retry **at most 3 times** and then stop to avoid infinite loops.

To recover missing data, call the "Re-pull Failed Callbacks" API manually.

Note: Developers must handle any business-level exceptions themselves. Once a callback has failed, repeating it is meaningless. Log the details and respond to the parking service with “processed” to terminate the retry cycle.

5. Signature-verification tips for callbacks

Some callback fields contain **boolean**, **float**, or **integer** values.

Do **not** convert them arbitrarily (e.g., amount `0.00` must remain `0.00` , not `0`), or the signature check will fail.

4. Query Interfaces (V2.0)

Query Zone Information

Brief Description

- This API is used to query zone information.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Query`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	Area	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules

Parameter	Location	Type	Required	Description	Notes
Data	body	object	Yes	Payload	Data object specific to this API
» PageIndex	body	integer	Yes	Page Index	Target page number (starts from 1 for ease of use)
» PageSize	body	integer	Yes	Page Size	Default: 20 records; Maximum: 100
» OrderBy	body	string	No	Sort Field	Field name to sort by; defaults to system-defined order if omitted
» OrderType	body	integer	Yes	Sort Order	0 = ASC (ascending), 1 = DESC (descending)
» QueryCondition	body	[object]	No	Query Conditions	Array of query condition objects
»» Field	body	string	Yes	Query Field	Name of the field to query
»» Operator	body	integer	Yes	Query Operator	Operator code; refer to "Query Operator Enum" for details
»» Value	body	string	Yes	Query Value	Value corresponding to the field type

Request Example

```

1  {
2      "AppId": "xb5glg0hgza40",
3      "Data": {
4          "PageIndex": 1,
5          "PageSize": 20,
6          "OrderBy": "Uid",
7          "OrderType": 0,
8          "QueryCondition": [
9
10         ]
11     },
12     "ParkKey": "1772492942362538",
13     "TimeStamp": "1774578368997",
14     "Nonce": "nfmpg55zpdg5",
15     "Sign": "rqvchzlkui1kws1jtinrltypgohs7idy6yp8tmfmcqy="
16 }

```

Return Parameter Description

Parameter	Type	Parameter Name	Required	Description	Remarks
ResultCode	integer		true	Request result code	For detailed information, please refer to the type

Parameter	Type	Parameter Name	Required	Description	Remarks
					enumeration/result code enumeration. 200 indicates successful operation by default.
ResultMsg	string		true	Request message	The information prompt returned by the API request.
ResultId	string		true	Request response identifier	An identifier for this specific processing response.
Data	object null		false	Specific data	The data object corresponding to the API response.
» PageIndex	integer		true	Current page number	The current page number, starting from 1 for easier comprehension.
» PageSize	integer		true	Page size	The number of items per page.
» TotalCount	integer		true	Total data count	The total number of data items.
» Results	[object] null		false	Current query data results	The results of the current data query.
»» Uid	integer		true	Unique primary key of the record	The unique primary key of the current record.
»» No	string		true	Record number	The unique number representing the current record.
»» AreaName	string		true	Area name	The name of the area.
»» AreaType	integer		true	Area type	Area type, 0 - Outer field, 1 - Nested inner field.
»» AreaSpaceNum	integer		true	Number of available parking spaces in the area	The count of available parking spaces in the area.
»» PUid	integer		true	Parent area primary key when nested	When the area is nested, this is the primary key of the associated parent area. For outer field areas, this value is 0.
»» Level	integer		true	Current nesting level of the area	The nesting level, indicating which layer the area belongs to.

Parameter	Type	Parameter Name	Required	Description	Remarks
					0 indicates the outermost outer field area.
»» CreationTime	string		true	Creation time	The timestamp when the data was created.

Return Example

```
{
  "Data": {
    "PageIndex": 1,
    "PageSize": 20,
    "TotalCount": 1,
    "Results": [
      {
        "Uid": 1,
        "No": "1772492942540835",
        "AreaName": "Outer Field",
        "AreaType": 0,
        "AreaSpaceNum": 500,
        "PUid": 0,
        "Level": 0,
        "CreationTime": "2026-03-18 02:32:14"
      }
    ]
  },
  "ResultCode": 200,
  "ResultMsg": "Query data successfully",
  "ResultId": "2037355457299906560"
}
```

Query Booth Information

Brief Description

- This API is used to query booth information.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Query`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	SentryHost	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» PageIndex	body	integer	Yes	Page Index	Target page number (starts from 1 for ease of use)
» PageSize	body	integer	Yes	Page Size	Default: 20 records; Maximum: 100
» OrderBy	body	string	No	Sort Field	Field name to sort by; defaults to system-defined order if omitted
» OrderType	body	integer	Yes	Sort Order	0 = ASC (ascending), 1 = DESC (descending)
» QueryCondition	body	[object]	No	Query Conditions	Array of query condition objects
»» Field	body	string	Yes	Query Field	Name of the field to query
»» Operator	body	integer	Yes	Query Operator	Operator code; refer to "Query Operator Enum" for details
»» Value	body	string	Yes	Query Value	Value corresponding to the field type

Request Example

```
1 {
2   "AppId": "xb5glg0hgza40",
3   "Data": {
4     "PageIndex": 1,
```

```

5      "PageSize": 20,
6      "OrderBy": "Uid",
7      "OrderType": 0,
8      "QueryCondition": []
9  },
10     "ParkKey": "1772492942362538",
11     "TimeStamp": "1774578631205",
12     "Nonce": "cfh6rc82o2xd",
13     "Sign": "neekq5zdw16a3bcn0owtfwxqo9nmher8q8wqbc+4sly="
14 }

```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.
» PageIndex	integer	true	Current page number	Page number starting from 1.
» PageSize	integer	true	Page size	Number of items per page.
» TotalCount	integer	true	Total data count	Total number of data items.
» Results	[object] null	false	Query results	Results of the current data query.
»» Uid	integer	true	Record primary key	Unique identifier for the record.
»» No	string	true	Record number	Unique number for the record.
»» Name	string	true	Sentry post name	Name of the sentry post.
»» Category	integer	true	Sentry post type	1 - Sentry post billing software, 2 - Sentry post background service.
»» IP	string	true	Sentry post IP address	IP address of the sentry post.
»» TcpPort	integer	true	Sentry post TCP communication port	TCP communication port of the sentry post.
»» BSPort	integer	true	Background port for B/S sentry post	Background port for B/S sentry post.
»» CreationTime	string	true	Creation time	Timestamp when the data was created.

Parameter	Type	Required	Description	Remarks
»» Longitude	number null	false	Sentry post longitude	Longitude of the sentry post.
»» Latitude	number null	false	Sentry post latitude	Latitude of the sentry post.

Return Example

```
{
  "Data": {
    "PageIndex": 1,
    "PageSize": 20,
    "TotalCount": 1,
    "Results": [
      {
        "Uid": 1,
        "No": "1772492942363478",
        "Name": "127.0.0.1",
        "Category": 1,
        "IP": "192.168.22.23",
        "TcpPort": null,
        "BSPort": 7702,
        "CreationTime": "2026-03-02 23:09:02",
        "Longitude": null,
        "Latitude": null
      }
    ]
  },
  "ResultCode": 200,
  "ResultMsg": "Query data successfully",
  "ResultId": "2037356557084164096"
}
```

Query Lane Information

Brief Description

- This API is used to query lane information.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Query`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	Lane	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
AppId	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» PageIndex	body	integer	Yes	Page Index	Target page number (starts from 1 for ease of use)
» PageSize	body	integer	Yes	Page Size	Default: 20 records; Maximum: 100
» OrderBy	body	string	No	Sort Field	Field name to sort by; defaults to system-defined order if omitted
» OrderType	body	integer	Yes	Sort Order	0 = ASC (ascending), 1 = DESC (descending)
» QueryCondition	body	[object]	No	Query Conditions	Array of query condition objects
»» Field	body	string	Yes	Query Field	Name of the field to query
»» Operator	body	integer	Yes	Query Operator	Operator code; refer to "Query Operator Enum" for details
»» Value	body	string	Yes	Query Value	Value corresponding to the field type

Request Example

```
1 {
2   "AppId": "xb5glg0hgzz40",
```

```

3      "Data": {
4          "PageIndex": 1,
5          "PageSize": 20,
6          "OrderBy": "Uid",
7          "OrderType": 0,
8          "QueryCondition": [
9
10         ]
11     },
12     "ParkKey": "1772492942362538",
13     "TimeStamp": "1774580431571",
14     "Nonce": "lmwjhyoe85x6",
15     "Sign": "s6gifh7clisx+rfpvqku57sfelrjb5skoikd+ryq9ky="
16 }

```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.
» PageIndex	integer	true	Current page number	Page number starting from 1.
» PageSize	integer	true	Page size	Number of items per page.
» TotalCount	integer	true	Total data count	Total number of data items.
» Results	[object] null	false	Query results	Results of the current data query.
»» Uid	integer	true	Record primary key	Unique identifier for the record.
»» No	string	true	Record number	Unique number for the record.
»» LaneName	string null	true	Lane name	Name of the lane.
»» LaneType	integer	true	Lane type	Lane type: 1 - Vehicle lane, 2 - Non-vehicle lane (pedestrian passage).
»» LaneDutyMode	integer	true	Lane duty mode	Lane duty mode: 0 - Unmanned lane, 1 - On-site duty.
»» CameraMode	integer	true	Camera mode	Camera mode: 1 - Single camera, 2 - Dual camera, 3 -

Parameter	Type	Required	Description	Remarks
				Multi-camera.
»» IdInterval	integer	true	Camera recognition interval	Recognition interval in seconds (effective for [dual camera mode, multi-camera mode]).
»» IsBackCar	integer null	true	Enable reverse car detection	Whether reverse car detection is enabled: 0 - Disabled, 1 - Enabled.
»» IsCharge	integer	true	Enable lane charging	Whether charging is enabled for the lane: 0 - Disabled, 1 - Enabled.
»» SentryHostNo	string	true	Sentry post number the lane belongs to	Number of the sentry post the lane belongs to.
»» CreationTime	string	true	Creation time	Timestamp when the data was created.
»» IsSense	integer	true	Enable ground loop judgment for license-plate-less vehicle QR code scanning	Enable ground loop judgment for license-plate-less vehicle QR code scanning: 0 - Disabled, 1 - Enabled.
»» IsSameInOut	integer	true	Same entry/exit mode	Whether it is the same entry/exit mode: 0 - No, 1 - Yes.
»» SameInOutLaneNo	string null	true	Associated lane number for same entry/exit mode	Associated lane number for same entry/exit mode.
»» SameInOutType	integer	true	Entry/exit type for lane in same entry/exit mode	Entry/exit type for lane in same entry/exit mode: 0 - Entry, 1 - Exit.
»» IsEnableBoard	integer	true	Enable lane control board	Whether lane control board is enabled: 0 - Disabled, 1 - Enabled.
»» LaneLinkAreas	[object] null	true	Area information bound to the current lane	Area information bound to the current lane.
»»» No	string	true	Record number	Unique number for the current record.
»»» AreaNo	string	true	Associated area number	Number of the associated area.
»»» InOutMode	integer	true	Entry/exit mode of the lane in the current area	Entry/exit mode of the lane in the current area: 0 - Exit lane, 1 - Entry lane.

Return Example

```
{
  "Data": {
    "PageIndex": 1,
    "PageSize": 20,
    "TotalCount": 2,
    "Results": [
      {
        "Uid": 1,
        "No": "1",
        "LaneName": "Entrance Lane",
        "LaneType": 1,
        "IdInterval": 3,
        "IsBackCar": 0,
        "IsCharge": 0,
        "OpenGateAndVoice": 1,
        "SentryHostNo": "1772492942363478",
        "CreationTime": "2026-03-02 23:09:02",
        "IsSense": 1,
        "IsSameInOut": 0,
        "SameInOutLaneNo": null,
        "SameInOutType": 0,
        "IsEnableBoard": 0,
        "LaneLinkAreas": [
          {
            "No": "1774609216714465",
            "AreaNo": "1772492942540835",
            "InOutMode": 1
          }
        ]
      },
      {
        "Uid": 2,
        "No": "2",
        "LaneName": "Exit Lane",
        "LaneType": 1,
        "IdInterval": 3,
        "IsBackCar": 0,
        "IsCharge": 1,
        "OpenGateAndVoice": 1,
        "SentryHostNo": "1772492942363478",
        "CreationTime": "2026-03-02 23:09:02",
        "IsSense": 1,
        "IsSameInOut": 0,
        "SameInOutLaneNo": null,
        "SameInOutType": 0,
        "IsEnableBoard": 0,
        "LaneLinkAreas": [
          {
            "No": "1774609216715573",
            "AreaNo": "1772492942540835",
            "InOutMode": 0
          }
        ]
      }
    ]
  },
  "ResultCode": 200,
  "ResultMsg": "Query data successfully",
}
```

```
"ResultId": "2037364113160568832"
}
```

Query Plate Type Information

Brief Description

- This API is used to query plate type information.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Query`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	CardType	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» PageIndex	body	integer	Yes	Page Index	Target page number (starts from 1 for ease of use)
» PageSize	body	integer	Yes	Page Size	Default: 20 records; Maximum: 100

Parameter	Location	Type	Required	Description	Notes
» OrderBy	body	string	No	Sort Field	Field name to sort by; defaults to system-defined order if omitted
» OrderType	body	integer	Yes	Sort Order	0 = ASC (ascending), 1 = DESC (descending)
» QueryCondition	body	[object]	No	Query Conditions	Array of query condition objects
»» Field	body	string	Yes	Query Field	Name of the field to query
»» Operator	body	integer	Yes	Query Operator	Operator code; refer to "Query Operator Enum" for details
»» Value	body	string	Yes	Query Value	Value corresponding to the field type

Request Example

```

1  {
2      "AppId": "xb5glg0hg40",
3      "Data": {
4          "PageIndex": 1,
5          "PageSize": 20,
6          "OrderBy": "Uid",
7          "OrderType": 0,
8          "QueryCondition": [
9
10         ]
11     },
12     "ParkKey": "1772492942362538",
13     "TimeStamp": "1774583643820",
14     "Nonce": "8pi08ligfp02",
15     "Sign": "mxzqjrpgn2ahlyqvckroopf4ivq/e0xjh9l4br1txay="
16 }

```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.

Parameter	Type	Required	Description	Remarks
Data	object null	false	Response data	Data object returned by the API.
» PageIndex	integer	true	Current page number	Page number starting from 1.
» PageSize	integer	true	Page size	Number of items per page.
» TotalCount	integer	true	Total data count	Total number of data items.
» Results	[object] null	false	Query results	Results of the current data query.
»» Uid	integer	true	Record primary key	Unique identifier for the record.
»» No	string	true	Record number	Unique number for the record.
»» ParkingCardTypeName	string null	true	Vehicle plate type name	Name of the vehicle plate type.
»» IsMoreCar	boolean	true	Whether used for multiple vehicles per parking space	Whether used for multiple vehicles per parking space.
»» CreationTime	string	true	Creation time	Timestamp when the data was created.
»» OperatorName	string null	true	Operator name	Name of the operator who performed the action.
»» Category	string	true	Vehicle plate classification	Vehicle plate classification: 3648, Temporary Vehicle D 3649, Temporary Vehicle C 3650, Temporary Vehicle B 3651, Temporary Vehicle A 3652, Monthly Vehicle A 3653, Monthly Vehicle B 3654, Monthly Vehicle C 3655, Monthly Vehicle D 3656, Free Vehicle 3657, Prepaid Vehicle

Return Example

```
{
  "Data": {
    "PageIndex": 1,
    "PageSize": 20,
    "TotalCount": 18,
    "Results": [
      {
        "Uid": 1,
        "No": "1772492942389944",
```

```
"ParkingCardTypeName": "Monthly Rental Car A",
"IsMoreCar": false,
"CreationTime": "2026-03-02 23:09:02",
"OperatorName": null,
"Category": "3652"
},
{
  "Uid": 2,
  "No": "1772492942389782",
  "ParkingCardTypeName": "Monthly Rental Car B",
  "IsMoreCar": true,
  "CreationTime": "2026-03-02 23:09:02",
  "OperatorName": null,
  "Category": "3653"
},
{
  "Uid": 3,
  "No": "1772492942389940",
  "ParkingCardTypeName": "Monthly Rental Car C",
  "IsMoreCar": false,
  "CreationTime": "2026-03-02 23:09:02",
  "OperatorName": null,
  "Category": "3654"
},
{
  "Uid": 4,
  "No": "1772492942389560",
  "ParkingCardTypeName": "Monthly Rental Car D",
  "IsMoreCar": false,
  "CreationTime": "2026-03-02 23:09:02",
  "OperatorName": null,
  "Category": "3655"
},
{
  "Uid": 5,
  "No": "1772492942389853",
  "ParkingCardTypeName": "Monthly Rental Car E",
  "IsMoreCar": false,
  "CreationTime": "2026-03-02 23:09:02",
  "OperatorName": null,
  "Category": "3661"
},
{
  "Uid": 6,
  "No": "1772492942389129",
  "ParkingCardTypeName": "Monthly Rental Car F",
  "IsMoreCar": false,
  "CreationTime": "2026-03-02 23:09:02",
  "OperatorName": null,
  "Category": "3662"
},
{
  "Uid": 7,
  "No": "1772492942389699",
  "ParkingCardTypeName": "Monthly Rental Car G",
  "IsMoreCar": false,
  "CreationTime": "2026-03-02 23:09:02",
  "OperatorName": null,
  "Category": "3663"
},
}
```

```
{
  "Uid": 8,
  "No": "1772492942389607",
  "ParkingCardTypeName": "Monthly Rental Car H",
  "IsMoreCar": false,
  "CreationTime": "2026-03-02 23:09:02",
  "OperatorName": null,
  "Category": "3664"
},
{
  "Uid": 9,
  "No": "1772492942389506",
  "ParkingCardTypeName": "Temporary Car A",
  "IsMoreCar": false,
  "CreationTime": "2026-03-02 23:09:02",
  "OperatorName": null,
  "Category": "3651"
},
{
  "Uid": 10,
  "No": "1772492942389238",
  "ParkingCardTypeName": "Temporary Car B",
  "IsMoreCar": false,
  "CreationTime": "2026-03-02 23:09:02",
  "OperatorName": null,
  "Category": "3650"
},
{
  "Uid": 11,
  "No": "1772492942389881",
  "ParkingCardTypeName": "Temporary Car C",
  "IsMoreCar": false,
  "CreationTime": "2026-03-02 23:09:02",
  "OperatorName": null,
  "Category": "3649"
},
{
  "Uid": 12,
  "No": "1772492942389922",
  "ParkingCardTypeName": "Temporary Car D",
  "IsMoreCar": false,
  "CreationTime": "2026-03-02 23:09:02",
  "OperatorName": null,
  "Category": "3648"
},
{
  "Uid": 13,
  "No": "1772492942389720",
  "ParkingCardTypeName": "Temporary Car E",
  "IsMoreCar": false,
  "CreationTime": "2026-03-02 23:09:02",
  "OperatorName": null,
  "Category": "3647"
},
{
  "Uid": 14,
  "No": "1772492942389279",
  "ParkingCardTypeName": "Temporary Car F",
  "IsMoreCar": false,
  "CreationTime": "2026-03-02 23:09:02",
```

```

        "OperatorName": null,
        "Category": "3646"
    },
    {
        "Uid": 15,
        "No": "1772492942389113",
        "ParkingCardTypeName": "Temporary Car G",
        "IsMoreCar": false,
        "CreationTime": "2026-03-02 23:09:02",
        "OperatorName": null,
        "Category": "3645"
    },
    {
        "Uid": 16,
        "No": "1772492942389629",
        "ParkingCardTypeName": "Temporary Car H",
        "IsMoreCar": false,
        "CreationTime": "2026-03-02 23:09:02",
        "OperatorName": null,
        "Category": "3644"
    },
    {
        "Uid": 17,
        "No": "1772492942389454",
        "ParkingCardTypeName": "Free Car",
        "IsMoreCar": false,
        "CreationTime": "2026-03-02 23:09:02",
        "OperatorName": null,
        "Category": "3656"
    },
    {
        "Uid": 18,
        "No": "1772492942389308",
        "ParkingCardTypeName": "Stored Value Car",
        "IsMoreCar": false,
        "CreationTime": "2026-03-02 23:09:02",
        "OperatorName": null,
        "Category": "3657"
    }
]
},
"ResultCode": 200,
"ResultMsg": "Query data successfully",
"ResultId": "2037377586011602944"
}

```

Query Plate Color Information

Brief Description

- This API is used to query plate information.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Query`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	CarType	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
ApplId	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» PageIndex	body	integer	Yes	Page Index	Target page number (starts from 1 for ease of use)
» PageSize	body	integer	Yes	Page Size	Default: 20 records; Maximum: 100
» OrderBy	body	string	No	Sort Field	Field name to sort by; defaults to system-defined order if omitted
» OrderType	body	integer	Yes	Sort Order	<code>0</code> = ASC (ascending), <code>1</code> = DESC (descending)
» QueryCondition	body	[object]	No	Query Conditions	Array of query condition objects
»» Field	body	string	Yes	Query Field	Name of the field to query
»»» Operator	body	integer	Yes	Query Operator	Operator code; refer to "Query Operator Enum" for details

Parameter	Location	Type	Required	Description	Notes
»» Value	body	string	Yes	Query Value	Value corresponding to the field type

Request Example

```

1  {
2    "AppId": "xb5glg0hg40",
3    "Data": {
4      "PageIndex": 1,
5      "PageSize": 20,
6      "OrderBy": "Uid",
7      "OrderType": 0,
8      "QueryCondition": []
9    },
10   "ParkKey": "1772492942362538",
11   "TimeStamp": "1774583183257",
12   "Nonce": "92an8dgzpc9n",
13   "Sign": "np8x2cbizdylxokvecrq1e0mhmxin4mitk6qc1glh5e="
14 }

```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.
» PageIndex	integer	true	Current page number	Page number starting from 1.
» PageSize	integer	true	Page size	Number of items per page.
» TotalCount	integer	true	Total data count	Total number of data items.
» Results	[object] null	false	Query results	Results of the current data query.
»» Uid	integer	true	Record primary key	Unique identifier for the record.
»» No	string	true	Record number	Unique number for the record.

Parameter	Type	Required	Description	Remarks
»» VehicleTypeName	string null	true	Vehicle type name	Name of the vehicle type.
»» CreationTime	string	true	Creation time	Timestamp when the data was created.
»» OperatorName	string null	true	Operator name	Name of the operator who performed the action.

Return Example

```
{
  "Data": {
    "PageIndex": 1,
    "PageSize": 20,
    "TotalCount": 9,
    "Results": [
      {
        "Uid": 1,
        "No": "1772492942395203",
        "VehicleTypeName": "Blue Plate Vehicle",
        "CreationTime": "2026-03-02 23:09:02",
        "OperatorName": null
      },
      {
        "Uid": 2,
        "No": "1772492942395558",
        "VehicleTypeName": "Yellow Plate Vehicle",
        "CreationTime": "2026-03-02 23:09:02",
        "OperatorName": null
      },
      {
        "Uid": 3,
        "No": "1772492942395297",
        "VehicleTypeName": "Green Plate Vehicle",
        "CreationTime": "2026-03-02 23:09:02",
        "OperatorName": null
      },
      {
        "Uid": 4,
        "No": "1772492942395388",
        "VehicleTypeName": "Black Plate Vehicle",
        "CreationTime": "2026-03-02 23:09:02",
        "OperatorName": null
      },
      {
        "Uid": 5,
        "No": "1772492942395917",
        "VehicleTypeName": "White Plate Vehicle",
        "CreationTime": "2026-03-02 23:09:02",
        "OperatorName": null
      },
      {
        "Uid": 6,
        "No": "1772492942395140",
        "VehicleTypeName": "No Plate Vehicle",
        "CreationTime": "2026-03-02 23:09:02",

```

```
    "OperatorName": null
  },
  {
    "Uid": 7,
    "No": "1772492942395720",
    "VehicleTypeName": "0 Plate Vehicle",
    "CreationTime": "2026-03-02 23:09:02",
    "OperatorName": null
  },
  {
    "Uid": 8,
    "No": "1772492942395498",
    "VehicleTypeName": "Yellow-Green Plate Vehicle",
    "CreationTime": "2026-03-02 23:09:02",
    "OperatorName": null
  },
  {
    "Uid": 9,
    "No": "1773111596539522",
    "VehicleTypeName": "test",
    "CreationTime": "2026-03-10 02:59:56",
    "OperatorName": null
  }
]
},
"ResultCode": 200,
"ResultMsg": "Query data successfully",
"ResultId": "2037375654295535616"
}
```

Query Lane Ground Loop Barrier Status

Brief Description

- This API is used to query the status of ground loop and barrier gates in a lane.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/GetLaneStatus`

Request Method

- GET

Request Content-Type

- `application/json`

Request Parameters

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier

Parameter	Location	Type	Required	Description	Notes
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» LaneNo	body	string	Yes	Lane number to query	Lane number to query

Request Example

```

1  {
2      "ParkKey": "1772492942362538",
3      "AppId": "xb5glg0hgZ40",
4      "TimeStamp": "1774591364770",
5      "Nonce": "389012",
6      "Sign": "fy23v3b2rlby2geg94sax3tge0j2fdd2hszwtzquf18=",
7      "Data": {
8          "LaneNo": "1"
9      }
10 }
```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.
» GataStatus	integer	true	Barrier gate status	Barrier gate status: 0 - Closed, 1 - Open, -1 - Unknown status
» GateLongOpen	integer	true	Whether barrier gate is set to stay open	Whether barrier gate is set to stay open: 0 - Disabled, 1 - Enabled, -1 - Unknown

Parameter	Type	Required	Description	Remarks
» GSStatus	integer	true	Ground loop status	Ground loop status: 0 - No detection, 1 - Detection active, -1 - Unknown status
» IDCameraStatus	integer	true	Main recognition camera status	Main recognition camera status: 0 - Device offline, 1 - Device online

Return Example

```
{
  "Data": {
    "GataStatus": -1,
    "GateLongOpen": 0,
    "GSStatus": -1,
    "IDCameraStatus": 0
  },
  "ResultCode": 200,
  "ResultMsg": "Lane ground loop gate status retrieved successfully!",
  "ResultId": "2037409969930797056"
}
```

Query Lane Current Order

Brief Description

- This API is used to query the current order of a lane.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/GetLaneOrder`

Request Method

- GET

Request Content-Type

- `application/json`

Request Parameters

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot

Parameter	Location	Type	Required	Description	Notes
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» LaneNo	body	string	Yes	Lane number to query	Lane number to query

Request Example

```

1  {
2      "ParkKey": "1772492942362538",
3      "AppId": "xb5glg0hgZ40",
4      "TimeStamp": "1774593426011",
5      "Nonce": "272946",
6      "Sign": "cmwzgjt3qj+thzq3myrdjqgzbng3qkrdid/kqriqlwo=",
7      "Data": {
8          "LaneNo": "2"
9      }
10 }
```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.
» OrderNo	string null	true	Parking order	Unique parking order created when the vehicle enters.
» CouponSchemeNos	[string]	true	Collection of used coupon codes	Collection of coupon codes that have been used.
» ChargeMoney	number	true	Current billing amount	Current billing amount.

Return Example

```
{
  "Data": {
    "OrderNo": "1774546385410449-162YRG",
    "ChargeMoney": 5.0,
    "CouponKey": []
  },
  "ResultCode": 200,
  "ResultMsg": "Lane order information retrieved successfully!",
  "ResultId": "2037418615238590464"
}
```

Query Vehicle Record

Brief Description

- This API is used to query vehicle records.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Query`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	MonthlyCar	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string

Parameter	Location	Type	Required	Description	Notes
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» PageIndex	body	integer	Yes	Page Index	Target page number (starts from 1 for ease of use)
» PageSize	body	integer	Yes	Page Size	Default: 20 records; Maximum: 100
» OrderBy	body	string	No	Sort Field	Field name to sort by; defaults to system-defined order if omitted
» OrderType	body	integer	Yes	Sort Order	0 = ASC (ascending), 1 = DESC (descending)
» QueryCondition	body	[object]	No	Query Conditions	Array of query condition objects
»» Field	body	string	Yes	Query Field	Name of the field to query
»» Operator	body	integer	Yes	Query Operator	Operator code; refer to "Query Operator Enum" for details
»» Value	body	string	Yes	Query Value	Value corresponding to the field type

Request Example

```

1  {
2      "AppId": "xb5glg0hgZ40",
3      "Data": {
4          "PageIndex": 1,
5          "PageSize": 20,
6          "OrderBy": "Uid",
7          "OrderType": 0,
8          "QueryCondition": [
9
10         ]
11     },
12     "ParkKey": "1772492942362538",
13     "TimeStamp": "1774593936520",
14     "Nonce": "17vkb8xj0u17",
15     "Sign": "poxjnf arjespzerzqole5tlf itetgn/hlhp3/kjveea="
16 }

```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.
» PageIndex	integer	true	Current page number	Page number starting from 1.
» PageSize	integer	true	Page size	Number of items per page.
» TotalCount	integer	true	Total data count	Total number of data items.
» Results	[object] null	false	Query results	Results of the current data query.
»» Uid	integer	true	Record primary key	Unique identifier for the record.
»» No	string	true	Record number	Unique number for the record.
»» LicensePlateNumber	string null	true	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default).
»» SystemSpotNumber	string	true	System parking space number	System parking space number.
»» ParkingSpotNumber	string null	true	Parking lot space number	Parking lot space number.
»» OwnerParkingSpaceCount	integer	true	Owner's parking space count	Number of parking spaces owned by the vehicle owner.
»» OwnerName	string null	true	Owner name	Name of the vehicle owner recorded in the issued vehicle record.
»» OwnerNo	string null	true	Owner number	Unique identifier number of the vehicle owner.
»» OwnerIDCard	string null	true	Owner ID card information	Owner ID card information (data needs to be decrypted).
»» OwnerLicense	string null	true	Owner driver's license	Owner driver's license (data needs to be decrypted).

Parameter	Type	Required	Description	Remarks
»» OwnerAddress	string null	true	Owner address	Residential address of the vehicle owner.
»» PhoneNumber	string	true	Phone number	Phone number (data needs to be decrypted).
»» Email	string null	true	Email address	Email address (data needs to be decrypted).
»» StartTime	string null	true	Start time	Start time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» EndTime	string null	true	End time	End time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» StoredVehicleBalance	number	true	Prepaid vehicle balance	Prepaid vehicle balance (in yuan, accurate to two decimal places).
»» VehicleTypeNo	string null	true	Vehicle type number	Vehicle type number.
»» VehicleTypeName	string null	true	Vehicle type name	Vehicle type name.
»» ParkingCardTypeNo	string null	true	License plate type number	License plate type number.
»» ParkingCardTypeName	string null	true	License plate type name	License plate type name.
»» OperatorName	string null	true	Operator name	Name of the operator.
»» CreationTime	string	true	Creation time	Creation time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» Remarks	string null	true	Remarks information	Remarks information.
»» Status	integer	true	License plate status	License plate status: 1 - Normal, 2 - Blacklist, 3 - Disabled.
»» EnableOffline	integer	true	Whitelist enable	Automatic gate opening when camera is offline (whitelist): 0 - Disabled, 1 - Enabled.

Parameter	Type	Required	Description	Remarks
»» StopSpaceList	[object]	true	Vehicle allowed parking area information	Information about areas where the vehicle is allowed to park.
»»» No	string	true	Record number	Unique number for the current record.
»»» Type	integer	true	Parking area type	Parking area type: 0 - Allowed in all areas, 1 - Allowed in specified areas.
»»» AreaNos	string	true	Allowed area number array	Array of allowed area numbers, e.g.: ["1692060223198448"].
»»» AreaName	string	true	Brief display of allowed area names	Brief display of allowed area names.
»»» SpaceCount	integer	true	Number of available parking spaces in the area	Number of available parking spaces in the area.

Return Example

```
{
  "Data": {
    "PageIndex": 1,
    "PageSize": 20,
    "TotalCount": 1,
    "Results": [
      {
        "Uid": 23,
        "No": "1774336597411515",
        "LicensePlateNumber": "DASH1",
        "SystemSpotNumber": "C00003",
        "ParkingSpotNumber": "",
        "OwnerParkingSpaceCount": 1,
        "OwnerName": "",
        "OwnerNo": "1774322212039836",
        "OwnerIDCard": "",
        "OwnerLicense": "",
        "OwnerAddress": "",
        "PhoneNumber": "",
        "Email": "",
        "StartTime": "2026-03-24 00:00:00",
        "EndTime": "2026-04-23 23:59:59",
        "StoredVehicleBalance": 0,
        "VehicleTypeNo": "1772492942395203",
        "VehicleTypeName": "Blue Plate Vehicle",
        "ParkingCardTypeNo": "1772492942389944",
        "ParkingCardTypeName": "Monthly Rental Car A",
        "OperatorName": "qq",
        "CreationTime": "2026-03-24 03:16:52",
```

```
    "Remarks": "",
    "Status": 1,
    "EnableOffline": 1,
    "IsDeleteOwner": false,
    "StopSpaceList": [
      {
        "No": "1774336597411678",
        "Type": 0,
        "AreaNos": "[\\"0\\"]",
        "AreaName": "All Zone",
        "SpaceCount": 1
      }
    ],
    "StatusText": "Normal"
  }
],
"ResultCode": 200,
"ResultMsg": "Query data successfully",
"ResultId": "2037420756380450816"
}
```

Query Blacklist Records

Brief Description

- This API is used to query blacklist records.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Query`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	BlackList	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier

Parameter	Location	Type	Required	Description	Notes
AppId	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» PageIndex	body	integer	Yes	Page Index	Target page number (starts from 1 for ease of use)
» PageSize	body	integer	Yes	Page Size	Default: 20 records; Maximum: 100
» OrderBy	body	string	No	Sort Field	Field name to sort by; defaults to system-defined order if omitted
» OrderType	body	integer	Yes	Sort Order	0 = ASC (ascending), 1 = DESC (descending)
» QueryCondition	body	[object]	No	Query Conditions	Array of query condition objects
»» Field	body	string	Yes	Query Field	Name of the field to query
»» Operator	body	integer	Yes	Query Operator	Operator code; refer to "Query Operator Enum" for details
»» Value	body	string	Yes	Query Value	Value corresponding to the field type

Request Example

```

1  {
2      "AppId": "xb5glg0hgZ40",
3      "Data": {
4          "PageIndex": 1,
5          "PageSize": 20,
6          "OrderBy": "Uid",
7          "OrderType": 0,
8          "QueryCondition": [
9
10         ]
11     },
12     "ParkKey": "1772492942362538",
13     "TimeStamp": "1774666337712",
14     "Nonce": "gzwnmt2f85kn",
15     "Sign": "94yrnyiz9kvdnozdfomnirgg+klb2mzgrmfocccsqhk="

```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.
» PageIndex	integer	true	Current page number	Page number starting from 1.
» PageSize	integer	true	Page size	Number of items per page.
» TotalCount	integer	true	Total data count	Total number of data items.
» Results	[object] null	false	Query results	Results of the current data query.
»» Uid	integer	true	Record primary key	Unique identifier for the record.
»» No	string	true	Record number	Unique number for the record.
»» LicensePlateNumber	string null	true	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default).
»» OwnerName	string null	true	Contact person name	Contact person name.
»» PhoneNumber	string	true	Phone number	Phone number.
»» StartTime	string null	true	Start time	Start time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» EndTime	string null	true	End time	End time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» Remarks	string null	true	Remarks information	Remarks information.

Parameter	Type	Required	Description	Remarks
»» CreationTime	string	true	Creation time	Creation time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» OperatorName	string null	true	Operator name	Name of the operator.
»» Status	integer	true	Blacklist status	Status value: 1 - Enabled, 0 - Disabled.

Return Example

```
{
  "Data": {
    "PageIndex": 1,
    "PageSize": 20,
    "TotalCount": 1,
    "Results": [
      {
        "Uid": 3,
        "No": "BL1774695130227262",
        "LicensePlateNumber": "ABC123",
        "OwnerName": "",
        "PhoneNumber": "",
        "StartTime": "2026-03-28 00:00:00",
        "EndTime": "2026-04-30 00:00:00",
        "Remarks": "1",
        "CreationTime": "2026-03-28 10:52:10",
        "OperatorName": "qq",
        "Status": 1
      }
    ]
  },
  "ResultCode": 200,
  "ResultMsg": "Query data successfully",
  "ResultId": "2037724429413285888"
}
```

Query Parking Order Records

Brief Description

- Query parking order records.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Query`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	ParkingOrder	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» PageIndex	body	integer	Yes	Page Index	Target page number (starts from 1 for ease of use)
» PageSize	body	integer	Yes	Page Size	Default: 20 records; Maximum: 100
» OrderBy	body	string	No	Sort Field	Field name to sort by; defaults to system-defined order if omitted
» OrderType	body	integer	Yes	Sort Order	<code>0</code> = ASC (ascending), <code>1</code> = DESC (descending)
» QueryCondition	body	[object]	No	Query Conditions	Array of query condition objects
»» Field	body	string	Yes	Query Field	Name of the field to query
»» Operator	body	integer	Yes	Query Operator	Operator code; refer to "Query Operator Enum" for details
»» Value	body	string	Yes	Query Value	Value corresponding to the field type

Request Example

```

1  {
2      "AppId": "xb5glg0hgZ40",
3      "Data": {
4          "PageIndex": 1,
5          "PageSize": 20,
6          "OrderBy": "Uid",
7          "OrderType": 0,
8          "QueryCondition": [
9
10         ]
11     },
12     "ParkKey": "1772492942362538",
13     "TimeStamp": "1774666807228",
14     "Nonce": "b4q1jiesvz40",
15     "Sign": "5uihmatmcmrzdekraeipvyaoragmflzKpbli1ezbj14="
16 }

```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object	false	Response data	Data object returned by the API.
» PageIndex	integer	true	Current page number	Page number starting from 1.
» PageSize	integer	true	Page size	Number of items per page.
» TotalCount	integer	true	Total data count	Total number of data items.
» Results	[object]	false	Query results	Results of the current data query.
»» Uid	integer	true	Record primary key	Unique identifier for the record.
»» OrderNo	string null	true	Parking order	Unique parking order created when the vehicle enters.
»» LicensePlateNumber	string null	true	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default).

Parameter	Type	Required	Description	Remarks
»» EntryTime	string null	true	Entry time	Entry time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» ExitTime	string null	true	Exit time	Exit time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» EntryLaneNo	string null	true	Entry lane number	Entry lane number.
»» EntryLaneName	string null	true	Entry lane name	Entry lane name.
»» ExitLaneNo	string null	true	Exit lane number	Exit lane number.
»» ExitLaneName	string null	true	Exit lane name	Exit lane name.
»» EntryOperatorName	string null	true	Entry operator	Entry operator name.
»» ExitOperatorName	string null	true	Exit operator	Exit operator name.
»» ParkingAreaNo	string	true	Parking area number	Current parking area number.
»» ParkingAreaName	string null	true	Parking area name	Current parking area name.
»» VehicleTypeNo	string null	true	Vehicle type number	Vehicle type number.
»» VehicleTypeName	string null	true	Vehicle type name	Vehicle type name.
»» EntryImageUrl	string null	true	Entry capture image URL	Entry passage capture image URL. If cloud platform is enabled with image storage, this is a cloud-stored image URL.
»» ExitImageUrl	string null	true	Exit capture image URL	Exit passage capture image URL. If cloud platform is enabled with image storage, this is a cloud-stored image URL.
»» ParkingCardTypeNo	string null	true	License plate type number	License plate type number.
»» ParkingCardTypeName	string null	true	License plate type name	License plate type name.

Parameter	Type	Required	Description	Remarks
»» OrderStatus	integer	true	Parking order status	For details, refer to the type enumeration/order status enumeration.
»» IsFreePass	boolean	true	Whether free pass is granted	Whether free pass is granted.
»» FreePassReason	string null	true	Free pass reason	Reason for granting free pass when vehicle is allowed free passage.
»» TotalAmountDue	number	true	Total amount due	Total amount due (in yuan, accurate to two decimal places).
»» TotalAmountPaid	number	true	Total amount paid	Total amount paid (in yuan, accurate to two decimal places).
»» IsVehicleLocked	boolean	true	Whether vehicle is locked	Whether vehicle is locked.
»» Remarks	string null	true	Remarks information	Remarks information.

Return Example

```
{
  "Data": {
    "PageIndex": 1,
    "PageSize": 20,
    "TotalCount": 1,
    "Results": [
      {
        "Uid": 140,
        "OrderNo": "1774695602882864-BCDE11",
        "LicensePlateNumber": "ABCDE11",
        "EntryTime": "2026-02-27 08:59:25",
        "ExitTime": "0001-01-01 00:00:00",
        "EntryLaneNo": "1",
        "EntryLaneName": "Entrance Lane",
        "ExitLaneNo": null,
        "ExitLaneName": null,
        "EntryOperatorName": "qq",
        "ExitOperatorName": null,
        "ParkingAreaNo": "1772492942540835",
        "ParkingAreaName": "Outer Field",
        "VehicleTypeNo": "1772492942395203",
        "VehicleTypeName": "Blue Plate Vehicle",
        "EntryImageUrl": "",
        "ExitImageUrl": null,
        "ParkingCardTypeNo": "1772492942389506",
        "ParkingCardTypeName": "Temporary Car A",
        "OrderStatus": 200,
        "IsFreePass": false,
        "FreePassReason": null,
      }
    ]
  }
}
```

```
        "TotalAmountDue": 0,
        "TotalAmountPaid": 0,
        "IsVehicleLocked": false,
        "Remarks": ""
    }
]
},
"ResultCode": 200,
"ResultMsg": "Query data successfully",
"ResultId": "2037726398689345536"
}
```

Query Payment Records

Brief Description

- This API is used to query payment records.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Query`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	PaymentOrder	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string

Parameter	Location	Type	Required	Description	Notes
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» PageIndex	body	integer	Yes	Page Index	Target page number (starts from 1 for ease of use)
» PageSize	body	integer	Yes	Page Size	Default: 20 records; Maximum: 100
» OrderBy	body	string	No	Sort Field	Field name to sort by; defaults to system-defined order if omitted
» OrderType	body	integer	Yes	Sort Order	0 = ASC (ascending), 1 = DESC (descending)
» QueryCondition	body	[object]	No	Query Conditions	Array of query condition objects
»» Field	body	string	Yes	Query Field	Name of the field to query
»» Operator	body	integer	Yes	Query Operator	Operator code; refer to "Query Operator Enum" for details
»» Value	body	string	Yes	Query Value	Value corresponding to the field type

Request Example

```

1  {
2    "AppId": "xb5glg0hgza40",
3    "Data": {
4      "PageIndex": 1,
5      "PageSize": 20,
6      "OrderBy": "Uid",
7      "OrderType": 0,
8      "QueryCondition": []
9    },
10   "ParkKey": "1772492942362538",
11   "TimeStamp": "1774667424134",
12   "Nonce": "gqxyppbbwp95d",
13   "Sign": "1a1t6tr1fdzbtboxnx2hk0cepz4of+3m7gx0i9q+c8="
14 }

```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by

Parameter	Type	Required	Description	Remarks
				default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.
» PageIndex	integer	true	Current page number	Page number starting from 1.
» PageSize	integer	true	Page size	Number of items per page.
» TotalCount	integer	true	Total data count	Total number of data items.
» Results	[object] null	false	Query results	Results of the current data query.
»» Uid	integer	true	Record primary key	Unique identifier for the record.
»» No	string	true	Record number	Unique number for the record.
»» OrderNo	string null	true	Parking order	Unique parking order created when the vehicle enters.
»» LicensePlateNumber	string null	true	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default).
»» OrderType	string	true	Order type	Order type (5902 - Monthly card recharge, 5901 - Temporary vehicle payment, 5903 - Wallet recharge, 5905 - Prepaid vehicle deduction, 5910 - Parking space renewal)
»» SystemSpotNumber	string null	true	System parking space number	System parking space number.
»» VehicleTypeNo	string null	true	Vehicle type number	Vehicle type number.
»» VehicleTypeName	string null	true	Vehicle type name	Vehicle type name.
»» ParkingCardTypeNo	string null	true	License plate type number	License plate type number.

Parameter	Type	Required	Description	Remarks
»» ParkingCardTypeName	string null	true	License plate type name	License plate type name.
»» StartTime	string null	true	Recharge start time	Recharge start time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» EndTime	string null	true	Recharge end time	Recharge end time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» BeforeEndTime	string null	true	End time before recharge	End time before recharge, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» OwnerNo	string null	true	Owner number	Unique identifier number of the vehicle owner.
»» OwnerName	string null	true	Owner name	Name of the vehicle owner recorded in the issued vehicle record.
»» TotalAmountDue	number	true	Total amount due	Total amount due (in yuan, accurate to two decimal places).
»» TotalAmountPaid	number	true	Total amount paid	Total amount paid (in yuan, accurate to two decimal places).
»» DiscountAmount	number	true	Discount amount	Discount amount (in yuan, accurate to two decimal places).
»» StoredValueDeduction	number	true	Prepaid value deduction	Prepaid value deduction.
»» SelfMoney	number	true	Self-service cash payment amount	Self-service cash payment amount.
»» OutReduceMoney	number	true	Change amount for self-service cash payment	Change amount for self-service cash payment.
»» PaymentMethod	string	true	Payment method number	For details, refer to the type enumeration/payment method enumeration information.
»» PaymentStatus	integer	true	Payment status	Payment status: 1 - Success, 0 - Unpaid, 2 - Failed, 3 - User canceled payment.
»» EntryTime	string null	true	Entry time	Entry time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).

Parameter	Type	Required	Description	Remarks
»» PaymentTime	string	true	Payment time	Payment time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» ParkingDuration	number	true	Parking duration	Parking duration (in minutes).
»» Remarks	string null	true	Remarks information	Remarks information.
»» OperatorName	string null	true	Operator name	Name of the operator.
»» CreationTime	string	true	Creation time	Creation time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» PaymentScene	integer	true	Payment scene	For details, refer to the type enumeration/payment scene enumeration.

Return Example

```
{
  "Data": {
    "PageIndex": 1,
    "PageSize": 20,
    "TotalCount": 1,
    "Results": [
      {
        "Uid": 33,
        "No": "MC1774322212074210",
        "OrderNo": "",
        "LicensePlateNumber": "DASH1",
        "OrderType": "5902",
        "SystemSpotNumber": "C00003",
        "VehicleTypeNo": "1772492942395203",
        "VehicleTypeName": "Blue Plate Vehicle",
        "ParkingCardTypeNo": "1772492942395203",
        "ParkingCardTypeName": "Monthly Rental Car A",
        "StartTime": "2026-04-23 23:59:59",
        "EndTime": "2026-03-24 00:00:00",
        "BeforeEndTime": null,
        "OwnerNo": "1774322212039836",
        "OwnerName": "",
        "TotalAmountDue": 0,
        "TotalAmountPaid": 0,
        "DiscountAmount": 0,
        "StoredValueDeduction": 0,
        "SelfMoney": 0,
        "OutReduceMoney": 0,
        "PaymentMethod": "79001",
        "PaymentStatus": 1,
        "EntryTime": null,
      }
    ]
  }
}
```

```
    "PaymentTime": "2026-03-24 03:16:52",
    "ParkingDuration": 0,
    "Remarks": "",
    "OperatorName": "qq",
    "CreationTime": "2026-03-24 03:16:52",
    "PaymentScene": 0
  }
],
},
"ResultCode": 200,
"ResultMsg": "Query data successfully",
"ResultId": "2037728986168066048"
}
```

Query Payment Detail Records

Brief Description

- This API is used to query payment detail records.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Query`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	PaymentOrderDetail	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string

Parameter	Location	Type	Required	Description	Notes
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» PageIndex	body	integer	Yes	Page Index	Target page number (starts from 1 for ease of use)
» PageSize	body	integer	Yes	Page Size	Default: 20 records; Maximum: 100
» OrderBy	body	string	No	Sort Field	Field name to sort by; defaults to system-defined order if omitted
» OrderType	body	integer	Yes	Sort Order	0 = ASC (ascending), 1 = DESC (descending)
» QueryCondition	body	[object]	No	Query Conditions	Array of query condition objects
»» Field	body	string	Yes	Query Field	Name of the field to query
»» Operator	body	integer	Yes	Query Operator	Operator code; refer to "Query Operator Enum" for details
»» Value	body	string	Yes	Query Value	Value corresponding to the field type

Request Example

```

1  {
2    "AppId": "xb5glg0hgza40",
3    "Data": {
4      "PageIndex": 1,
5      "PageSize": 20,
6      "OrderBy": "Uid",
7      "OrderType": 0,
8      "QueryCondition": []
9    },
10   "ParkKey": "1772492942362538",
11   "TimeStamp": "1774668088594",
12   "Nonce": "d8b74lmheaiz",
13   "Sign": "4kwgodq4slly1w6nsibfve7ln1vxvzqwaec2gzmgiiu="
14 }

```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by

Parameter	Type	Required	Description	Remarks
				default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.
» PageIndex	integer	true	Current page number	Page number starting from 1.
» PageSize	integer	true	Page size	Number of items per page.
» TotalCount	integer	true	Total data count	Total number of data items.
» Results	[object] null	false	Query results	Results of the current data query.
»» Uid	integer	true	Record primary key	Unique identifier for the record.
»» No	string	true	Record number	Unique number for the record.
»» OrderNo	string null	true	Parking order	Unique parking order created when the vehicle enters.
»» PaymentOrderNo	string	true	Payment order number	Payment order number.
»» OrderDetailNo	string null	true	Parking detail number	Parking detail number.
»» StartTime	string null	true	Billing start time	Billing start time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» EndTime	string null	true	Billing end time	Billing end time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» TimeCount	integer null	true	Time period	Time period (in minutes).
»» OrderType	string	true	Order type	Order type (5902 - Monthly card recharge, 5901 - Temporary vehicle payment, 5903 - Wallet recharge, 5905 -

Parameter	Type	Required	Description	Remarks
				Prepaid vehicle deduction, 5910 - Parking space renewal)
»» AreaNo	string null	true	Parking area number	Parking area number.
»» AreaName	string null	true	Parking area name	Parking area name.
»» LicensePlateNumber	string null	true	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default).
»» VehicleTypeNo	string null	true	Vehicle type number	Vehicle type number.
»» VehicleTypeName	string null	true	Vehicle type name	Vehicle type name.
»» ParkingCardTypeNo	string null	true	License plate type number	License plate type number.
»» ParkingCardTypeName	string null	true	License plate type name	License plate type name.
»» PaymentMethod	string	true	Payment method number	For details, refer to the type enumeration/payment method enumeration information.
»» TotalAmountDue	number	true	Total amount due	Total amount due (in yuan, accurate to two decimal places).
»» TotalAmountPaid	number	true	Total amount paid	Total amount paid (in yuan, accurate to two decimal places).
»» DiscountAmount	number	true	Discount amount	Discount amount (in yuan, accurate to two decimal places).
»» PaymentStatus	integer	true	Payment status	Payment status: 1 - Success, 0 - Unpaid, 2 - Failed, 3 - User canceled payment.
»» PassageLaneNo	string null	true	Passage lane number	Passage lane number.
»» Remarks	string null	true	Remarks information	Remarks information.
»» CreationTime	string	true	Creation time	Creation time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» OperatorName	string null	true	Operator name	Name of the operator.

Parameter	Type	Required	Description	Remarks
»» StoredValueDeduction	number	true	Prepaid value deduction	Prepaid value deduction.

Return Example

```

{
  "Data": {
    "PageIndex": 1,
    "PageSize": 20,
    "TotalCount": 1,
    "Results": [
      {
        "Uid": 33,
        "No": "PD1774322212119978-ASH1",
        "PaymentOrderNo": "MC1774322212074210",
        "OrderNo": "MC1774322212074210",
        "OrderDetailNo": null,
        "StartTime": null,
        "EndTime": "2026-03-24 03:16:52",
        "TimeCount": 0,
        "OrderType": "5902",
        "AreaNo": "",
        "AreaName": null,
        "LicensePlateNumber": "DASH1",
        "ParkingCardTypeNo": "1772492942389944",
        "ParkingCardTypeName": "Monthly Rental Car A",
        "VehicleTypeNo": "1772492942395203",
        "VehicleTypeName": "Blue Plate Vehicle",
        "PaymentMethod": "79001",
        "TotalAmountDue": 0,
        "DiscountAmount": 0,
        "TotalAmountPaid": 0,
        "PaymentStatus": 1,
        "PassageLaneNo": null,
        "Remarks": null,
        "CreationTime": "2026-03-24 03:16:52",
        "OperatorName": "qq",
        "StoredValueDeduction": 0
      }
    ]
  },
  "ResultCode": 200,
  "ResultMsg": "Query data successfully",
  "ResultId": "2037731773102718976"
}

```

Query Gate Release Record

Brief Description

- This API is used to query gate release records.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Query`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	OpenGate	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» PageIndex	body	integer	Yes	Page Index	Target page number (starts from 1 for ease of use)
» PageSize	body	integer	Yes	Page Size	Default: 20 records; Maximum: 100
» OrderBy	body	string	No	Sort Field	Field name to sort by; defaults to system-defined order if omitted
» OrderType	body	integer	Yes	Sort Order	<code>0</code> = ASC (ascending), <code>1</code> = DESC (descending)
» QueryCondition	body	[object]	No	Query Conditions	Array of query condition objects
»» Field	body	string	Yes	Query Field	Name of the field to query

Parameter	Location	Type	Required	Description	Notes
»» Operator	body	integer	Yes	Query Operator	Operator code; refer to "Query Operator Enum" for details
»» Value	body	string	Yes	Query Value	Value corresponding to the field type

Request Example

```

1  {
2    "AppId": "xb5glg0hgZ40",
3    "Data": {
4      "PageIndex": 1,
5      "PageSize": 20,
6      "OrderBy": "Uid",
7      "OrderType": 0,
8      "QueryCondition": []
9    },
10   "ParkKey": "1772492942362538",
11   "TimeStamp": "1774668802564",
12   "Nonce": "bcpf88dr2zvx",
13   "Sign": "yjqgppa5xtzw/gu5yhyy+rhhtv/wvzjitz8eytzwpls="
14 }

```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object	false	Response data	Data object returned by the API.
» PageIndex	integer	true	Current page number	Page number starting from 1.
» PageSize	integer	true	Page size	Number of items per page.
» TotalCount	integer	true	Total data count	Total number of data items.
» Results	[object]	true	Query results	Results of the current data query.
»» Uid	integer	true	Record primary key	Unique identifier for the record.

Parameter	Type	Required	Description	Remarks
»» No	string	true	Record number	Unique number for the record.
»» GateSwitchType	integer	true	Manual gate switch type	For details, refer to the type enumeration/manual gate switch type enumeration.
»» OrderNo	string null	true	Parking order	Unique parking order created when the vehicle enters.
»» LicensePlateNumber	string null	true	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default).
»» PassageLaneNo	string	true	Passage lane number	Passage lane number.
»» PassageLaneName	string	true	Passage lane name	Passage lane name.
»» DeviceNo	string null	true	Device number	Device number.
»» DeviceName	string null	true	Device name	Device name.
»» VehicleTypeNo	string null	true	Vehicle type number	Vehicle type number.
»» VehicleTypeName	string null	true	Vehicle type name	Vehicle type name.
»» ParkingCardTypeNo	string null	true	License plate type number	License plate type number.
»» ParkingCardTypeName	string null	true	License plate type name	License plate type name.
»» SpecialVehicleNo	string null	true	Special vehicle number	Special vehicle number.
»» SpecialVehicleName	string null	true	Special vehicle name	Special vehicle name.
»» PassageImageUrl	string	true	Passage capture image URL	Passage capture image URL (this is a local LAN HTTP address).
»» PassageTime	string	true	Passage time	Passage time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).

Parameter	Type	Required	Description	Remarks
»» OperatorName	string null	true	Operator name	Name of the operator.
»» OperationTime	string	true	Operation time	Operation time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» Remarks	string null	true	Remarks information	Remarks information.
»» TotalAmountPaid	number	true	Total amount paid	Total amount collected after license-plate-less vehicle exit.

Return Example

```
{
  "Data": {
    "PageIndex": 1,
    "PageSize": 20,
    "TotalCount": 1,
    "Results": [
      {
        "Uid": 1,
        "No": "ER1774697600304463",
        "GateSwitchType": 1,
        "OrderNo": null,
        "LicensePlateNumber": null,
        "PassageLaneNo": "2",
        "PassageLaneName": "Exit Lane",
        "DeviceNo": "1772492942377879",
        "DeviceName": "Exit Camera",
        "VehicleTypeNo": null,
        "VehicleTypeName": null,
        "ParkingCardTypeNo": "1772492942389506",
        "ParkingCardTypeName": "Temporary Car A",
        "SpecialVehicleNo": "",
        "SpecialVehicleName": "",
        "PassageImageUrl":
"http://192.168.22.23:8888/CameraCaptures//1772492942363478/202603/28/dfc7b20adf202603281133191563
007_Online.jpg",
        "PassageTime": "2026-03-28 11:33:20",
        "OperatorName": "qq",
        "OperationTime": "2026-03-28 11:33:20",
        "Remarks": "[Exit Camera]Release gate successfully!",
        "TotalAmountPaid": 0
      }
    ]
  },
  "ResultCode": 200,
  "ResultMsg": "Query data successfully",
  "ResultId": "2037734767680585728"
}
```

Query Coupon Scheme

Brief Description

- This API is used to query coupon schemes.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Query`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	CouponScheme	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» PageIndex	body	integer	Yes	Page Index	Target page number (starts from 1 for ease of use)
» PageSize	body	integer	Yes	Page Size	Default: 20 records; Maximum: 100
» OrderBy	body	string	No	Sort Field	Field name to sort by; defaults to system-defined order if omitted

Parameter	Location	Type	Required	Description	Notes
» OrderType	body	integer	Yes	Sort Order	0 = ASC (ascending), 1 = DESC (descending)
» QueryCondition	body	[object]	No	Query Conditions	Array of query condition objects
»» Field	body	string	Yes	Query Field	Name of the field to query
»» Operator	body	integer	Yes	Query Operator	Operator code; refer to "Query Operator Enum" for details
»» Value	body	string	Yes	Query Value	Value corresponding to the field type

Request Example

```

1  {
2    "AppId": "xb5glg0hg40",
3    "Data": {
4      "PageIndex": 1,
5      "PageSize": 20,
6      "OrderBy": "Uid",
7      "OrderType": 0,
8      "QueryCondition": []
9    },
10   "ParkKey": "1772492942362538",
11   "TimeStamp": "1774669296136",
12   "Nonce": "a33ys6g6sspg",
13   "Sign": "tejosus65710mv720egz4ztpug3ai2zlgbpqapdiwama="
14 }

```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.
» PageIndex	integer	true	Current page number	Page number starting from 1.
» PageSize	integer	true	Page size	Number of items per page.

Parameter	Type	Required	Description	Remarks
» TotalCount	integer	true	Total data count	Total number of data items.
» Results	[object] null	false	Query results	Results of the current data query.
»» Uid	integer	true	Record primary key	Unique identifier for the record.
»» No	string	true	Record number	Unique number for the record.
»» Name	string	true	Coupon scheme name	Coupon scheme name.
»» Type	string	true	Coupon discount type	Coupon discount type (101 - Discount amount, 102 - Discount duration, 103 - Discount ratio, 104 - Free until specified time)
»» Amount	number	true	Discount amount (for amount type)	Discount amount for amount type (unit: yuan).
»» Duration	number	true	Discount duration (for duration type)	Discount duration for duration type (unit: minutes).
»» Ratio	number	true	Discount ratio (for ratio type)	Discount ratio for ratio type.
»» VaildTime	string null	true	Free until specified time	Free until specified time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» Scene	integer	true	Coupon scheme usage scene	Coupon scheme usage scene: 0 - Only for in-field discount, 1 - Only for exit discount, 2 - Exit and in-field discount.
»» CreationTime	string	true	Creation time	Creation time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» OperatorName	string null	true	Operator name	Name of the operator.

Return Example

```
{
  "Data": {
    "PageIndex": 1,
    "PageSize": 20,
    "TotalCount": 1,
    "Results": [
      {
        "Uid": 9,
        "No": "1774060298299594",
```

```
    "Name": "forfree",
    "Type": "103",
    "Amount": 0,
    "Duration": 0,
    "Ratio": 0,
    "VaildTime": null,
    "Enable": null,
    "Scene": 2,
    "CreationTime": "2026-03-21 02:31:38",
    "OperatorName": "admin"
  }
],
"ResultCode": 200,
"ResultMsg": "Query data successfully",
"ResultId": "2037736837838045184"
}
```

Query Plate Coupon Records

Brief Description

- This API is used to query plate coupon records.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Query`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	PlateCoupon	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot

Parameter	Location	Type	Required	Description	Notes
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» PageIndex	body	integer	Yes	Page Index	Target page number (starts from 1 for ease of use)
» PageSize	body	integer	Yes	Page Size	Default: 20 records; Maximum: 100
» OrderBy	body	string	No	Sort Field	Field name to sort by; defaults to system-defined order if omitted
» OrderType	body	integer	Yes	Sort Order	0 = ASC (ascending), 1 = DESC (descending)
» QueryCondition	body	[object]	No	Query Conditions	Array of query condition objects
»» Field	body	string	Yes	Query Field	Name of the field to query
»» Operator	body	integer	Yes	Query Operator	Operator code; refer to "Query Operator Enum" for details
»» Value	body	string	Yes	Query Value	Value corresponding to the field type

Request Example

```
1  {
2    "AppId": "xb5glg0hg240",
3    "Data": {
4      "PageIndex": 1,
5      "PageSize": 20,
6      "OrderBy": "Uid",
7      "OrderType": 0,
8      "QueryCondition": []
9    },
10   "ParkKey": "1772492942362538",
11   "TimeStamp": "1774669664385",
12   "Nonce": "vafyomon72w1",
13   "Sign": "bgcg28vktlpata+lm1znvrfqz96o8htcefa+gprcsk="
14 }
```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.
» PageIndex	integer	true	Current page number	Page number starting from 1.
» PageSize	integer	true	Page size	Number of items per page.
» TotalCount	integer	true	Total data count	Total number of data items.
» Results	[object] null	false	Query results	Results of the current data query.
»» Uid	integer	true	Record primary key	Unique identifier for the record.
»» No	string	true	Record number	Unique number for the record.
»» CouponSchemeNo	string	true	Coupon scheme number	Coupon scheme number.
»» Type	string	true	Coupon discount type	Coupon discount type (101 - Discount amount, 102 - Discount duration, 103 - Discount ratio, 104 - Free until specified time)
»» Value	number	true	Discount value	For discount amount: the amount in yuan; for discount duration: the duration in minutes; for discount ratio: the discount ratio (greater than 0 and less than 10).
»» StartTime	string null	true	Start time	Start time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» EndTime	string null	true	End time	End time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).

Parameter	Type	Required	Description	Remarks
»» OrderNo	string null	true	Parking order	Unique parking order created when the vehicle enters.
»» LicensePlateNumber	string null	true	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default).
»» BusinessName	string null	true	Merchant name	Merchant name.
»» CreationTime	string	false	Creation time	Creation time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» Remarks	string null	true	Remarks information	Remarks information.

Return Example

```
{
  "Data": {
    "PageIndex": 1,
    "PageSize": 20,
    "TotalCount": 1,
    "Results": [
      {
        "Uid": 8,
        "No": "CF1774698341290726",
        "CouponSchemeNo": "1774060298299594",
        "Type": "103",
        "Value": 0,
        "StartTime": null,
        "EndTime": null,
        "OrderNo": "1774695602882864-BCDE11",
        "LicensePlateNumber": "ABCDE11",
        "BusinessName": null,
        "CreationTime": "2026-03-28 11:45:41",
        "Remarks": null
      }
    ]
  },
  "ResultCode": 200,
  "ResultMsg": "Query data successfully",
  "ResultId": "2037738382377910272"
}
```

Query Parking Fee for Specified Order

Brief Description

- This API is used to query the current parking fee based on the parking order.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/GetOrderPrice`

Request Method

- GET

Request Content-Type

- `application/json`

Request Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
AppId	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» OrderNo	body	string	Yes	Parking order to query	Parking order to query
» LicensePlateNumber	body	string	No	License plate number	License plate number
» PlateCouponNo	body	string	No	Coupon code	Coupon code

Request Example

```
1  {
2      "ParkKey": "1772492942362538",
3      "AppId": "xb5glg0hgz40",
4      "TimeStamp": "1774670142512",
5      "Nonce": "495296",
6      "Sign": "+94xw4q0vudgx4wit4elyz30z0erz8g0y0gf/8sy1ys=",
7      "Data": {
8          "OrderNo": "1774695602882864-BCDE11"
9      }
10 }
```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.
» OrderNo	string null	true	Parking order	Unique parking order created when the vehicle enters.
» CouponSchemeNos	[string]	true	Collection of used coupon codes	Collection of used coupon codes.
» LicensePlateNumber	number	true	License plate number	License plate number.
» EntryTime	string	true	Entry time	Entry time.
» CalcTime	string	true	Billing time	Billing time.
» TotalAmountDue	string	true	Total amount due	Total amount due.
» TotalAmountPaid	string	true	Total amount paid	Total amount paid.
» DiscountAmount	string	true	Discount amount	Discount amount.
» CalcDetail	string	true	Billing details	Billing details.
»» AreaNo	string	true	Area code	Area code.
»» AreaName	string	true	Area name	Area name.
»» StartTime	string	true	Start time	Start time.
»» EndTime	string	true	End time	End time.
»» CalcAmount	string	true	Billing amount	Billing amount.
»» FreeMinutes	string	true	Free minutes	Free minutes.
»» StayMinutes	string	true	Parking duration	Parking duration (in minutes).
»» Remarks	string	true	Remarks	Remarks.
»» IsOverTime	string	true	Whether overtime	Whether overtime (0 - No, 1 - Yes).
»» IsCarExpire	string	true	Whether expired	Whether expired (0 - No, 1 - Yes).

Return Example

```
{
  "Data": {
    "OrderNo": "1774695602882864-BCDE11",
    "LicensePlateNumber": "ABCDE11",
    "EntryTime": "2026-02-27 08:59:25",
    "CalcTime": "2026-03-28 11:55:42",
    "TotalAmountDue": 5.0,
    "TotalAmountPaid": 5.0,
    "DiscountAmount": 0.0,
    "CalcDetail": [
      {
        "AreaNo": "1772492942540835",
        "AreaName": "Outer Field",
        "StartTime": "2026-02-27 08:59:25",
        "EndTime": "2026-03-28 11:55:42",
        "CalcAmount": 5.0,
        "DiscountAmount": null,
        "FreeMinutes": 0,
        "StayMinutes": 41936,
        "Remarks": "Charged by Temporary Car A",
        "IsOverTime": 0,
        "IsCarExpire": 0
      }
    ],
    "PlateCouponNos": [
      {
        "No": "CF1774698341290726",
        "Name": "0.00% off"
      }
    ]
  },
  "ResultCode": 200,
  "ResultMsg": "Successfully retrieved parking fee information for the specified parking order!",
  "ResultId": "2037740387880173568"
}
```

5. Notification Interfaces (V2.0)

In-Parking Record Change Notification (Including Modify and Close)

Brief Description

- This API is used to manage modifications to in-parking records, including license plate number, license plate type, vehicle type, entry time, order closure, and order remarks.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Notice`

Request Method

- POST

Request Content-Type

- application/json

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	ModifyParkingOrder	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» NoticeType	body	integer	Yes	Notification type	Notification type: 0 - Update, 1 - Delete. Note: Delete operation here refers to closing in-field orders.
» Items	body	[object]	Yes	Notification data collection	Collection of notification data
»» OrderNo	body	string null	Yes	Parking order	Unique parking order created when the vehicle enters.
»» LicensePlateNumber	body	string null	Yes	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default).
»» VehicleTypeNo	body	string null	Yes	Vehicle type number	Vehicle type number.

Parameter	Location	Type	Required	Description	Notes
»» ParkingCardTypeNo	body	string null	Yes	License plate type number	License plate type number.
»» EntryTime	body	string null	Yes	Entry time	Entry time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» Remarks	body	string null	Yes	Remarks information	Remarks information.

Request Example

```
1  {
2    "AppId": "xb5glg0hg40",
3    "Data": {
4      "NoticeType": 0,
5      "Items": [
6        {
7          "OrderNo": "1774695602882864-BCDE11",
8          "EntryTime": "2026-03-28 14:01:53",
9          "Remarks": "test"
10       }
11     ]
12   },
13   "ParkKey": "1772492942362538",
14   "TimeStamp": "1774677754883",
15   "Nonce": "fve3txy88g8d",
16   "Sign": "3cnpkchws2g/hfa896vrusz82omxcqpmr4ckxm//fpge="
17 }
```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.

Return Example

```
{
  "ResultCode": 200,
  "ResultMsg": "Order [1774695602882864-BCDE11] on-site record updated successfully",
```

```
"ResultId": "2037772316146958336"
}
```

Notify Vehicle Changes (Including Add, Edit, Delete)

Brief Description

- This API is used to manage the addition, modification and deletion of vehicle records.
- Note: In the parking system, vehicles belonging to the same owner share the same start time and end time. Only one vehicle type is allowed per owner. If multiple vehicle records for the same owner are submitted during create or update operations, the record with the latest end time shall be used as the reference to update the database.

Note: In the current version, fixed vehicles are granted access to all zones by default. No API is available to modify this configuration.

Note

- 1.The combination of No, ParkingSpotNumber and OwnerNo is unique.If all three fields match an existing record, an update will be performed; otherwise, a new record will be created.
- 2.For the one-owner-multiple-vehicles scenario:If two license plates are uploaded in the initial creation, and a third plate is submitted later, the system will update the first license plate instead of adding a new record.
- 3.For the one-owner-multiple-vehicles scenario:If two license plates are registered initially, and a third license plate is included in the second request, the system will process all three plates without updating existing records.
- 4.If a vehicle that has already entered the parking lot is deregistered via this API, the vehicle will be converted to a temporary vehicle.Re-registration will close the previous parking order.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Notice`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	MonthlyCar	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object null	Yes	Response data	Data object returned by the API.
» NoticeType	body	integer	Yes	Notification type	Notification type: 0 - Add or Update, 1 - Delete.
» Items	body	[object]	Yes	Notification data collection	Collection of notification data
»» No	body	string	Yes	Record number	Unique number for the current record.
»» LicensePlateNumber	body	string null	Yes	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default).
»» ParkingSpotNumber	body	string null	Yes	Parking lot space number	Parking lot space number.
»» OwnerParkingSpaceCount	body	integer	Yes	Owner's parking space count	Number of parking spaces owned by the vehicle owner.
»» OwnerName	body	string null	Yes	Owner name	Name of the vehicle owner recorded in the issued vehicle record.
»» OwnerNo	body	string null	Yes	Owner number	Unique identifier number of the vehicle owner.

Parameter	Location	Type	Required	Description	Notes
»» OwnerIDCard	body	string null	Yes	Owner ID card information	Owner ID card information.
»» OwnerLicense	body	string null	Yes	Owner driver's license	Owner driver's license.
»» OwnerAddress	body	string	Yes	Owner address	Residential address of the vehicle owner.
»» PhoneNumber	body	string	Yes	Phone number	Phone number.
»» Email	body	string null	Yes	Email address	Email address.
»» StartTime	body	string null	Yes	Start time	Start time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» EndTime	body	string null	Yes	End time	End time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» StoredVehicleBalance	body	number	Yes	Prepaid vehicle balance	Prepaid vehicle balance (in yuan, accurate to two decimal places).
»» VehicleTypeNo	body	string null	Yes	Vehicle type number	Vehicle type number.
»» ParkingCardTypeNo	body	string null	Yes	License plate type number	License plate type number.
»» OperatorName	body	string null	Yes	Operator name	Name of the operator.
»» Remarks	body	string null	Yes	Remarks information	Remarks information.
»» EnableOffline	body	integer	Yes	Whitelist enable	Automatic gate opening when camera is offline (whitelist): 0 - Disabled, 1 - Enabled.

Parameter	Location	Type	Required	Description	Notes
»» IsDeleteOwner	body	boolean	Yes	Whether to delete owner information	When performing delete operation, whether to also delete owner information. Note: deleting owner information will also delete all associated vehicle information.

Request Example

```
1  {
2    "AppId": "xb5glg0hgZ40",
3    "Data": {
4      "NoticeType": 0,
5      "Items": [
6        {
7          "No": "630000201108254378",
8          "LicensePlateNumber": "P12345",
9          "ParkingSpotNumber": "56",
10         "OwnerParkingSpaceCount": 2,
11         "OwnerName": "ces",
12         "OwnerNo": "410000199701311192",
13         "OwnerIDCard": "",
14         "OwnerLicense": "test",
15         "OwnerAddress": "",
16         "PhoneNumber": "",
17         "Email": "",
18         "StartTime": "2024-02-27 11:04:31",
19         "EndTime": "2026-12-27 11:04:31",
20         "StoredVehicleBalance": 0,
21         "VehicleTypeNo": "1772492942395203",
22         "ParkingCardTypeNo": "1772492942389944",
23         "OperatorName": "",
24         "Remarks": "",
25         "EnableOffline": 1,
26         "IsDeleteOwner": false
27       }
28     ]
29   },
30   "ParkKey": "1772492942362538",
31   "TimeStamp": "1774678624879",
32   "Nonce": "1t3ypa7wchrn",
33   "Sign": "pkuxyeck2mm8v4na9voqee+il3mo5xdvbjrjwt34qe4="
34 }
```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.

Parameter	Type	Required	Description	Remarks
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.

Return Example

```
{
  "ResultCode": 200,
  "ResultMsg": "Add or update operation successfully!",
  "ResultId": "2037775965145300992"
}
```

Notify Blacklist Changes (Including Add, Edit, Delete)

Brief Description

- This API is used to manage the addition, update, and deletion of blacklist records.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Notice`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	BlackList	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier

Parameter	Location	Type	Required	Description	Notes
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» NoticeType	body	integer	Yes	Notification type	Notification type: 0 - Add or Update, 1 - Delete.
» Items	body	[object]	Yes	Notification data collection	Collection of notification data
»» No	body	string	Yes	Record number	Unique number for the current record.
»» LicensePlateNumber	body	string null	Yes	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default).
»» OwnerName	body	string null	Yes	Owner name	Name of the vehicle owner recorded in the blacklist record.
»» PhoneNumber	body	string null	Yes	Phone number	Phone number.
»» StartTime	body	string null	Yes	Start time	Start time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» EndTime	body	string null	Yes	End time	End time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» Remarks	body	string null	Yes	Remarks information	Remarks information.
»» CreationTime	body	string null	Yes	Creation time	Creation time, following the date format in basic rules,

Parameter	Location	Type	Required	Description	Notes
					accurate to seconds (e.g., 2023-08-01 09:34:51).
»» OperatorName	body	string null	Yes	Operator name	Name of the operator.

Request Example

```
1  {
2    "AppId": "xb5glg0hg40",
3    "Data": {
4      "NoticeType": 0,
5      "Items": [
6        {
7          "No": "1234567890",
8          "LicensePlateNumber": "B12345",
9          "OwnerName": "",
10         "PhoneNumber": "",
11         "StartTime": "2026-01-28 14:39:10",
12         "EndTime": "2026-03-28 14:39:10",
13         "Remarks": "",
14         "CreationTime": "2026-03-28 14:39:10",
15         "OperatorName": ""
16       }
17     ]
18   },
19   "ParkKey": "1772492942362538",
20   "TimeStamp": "1774679966590",
21   "Nonce": "ofmdq0shb7xr",
22   "Sign": "j7tr3+laeqxgc7ubypekl95tzgzct+f8ut4qsmsb7ik="
23 }
```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.

Return Example

```
{
  "ResultCode": 200,
  "ResultMsg": "Add or update operation successfully!",
```

```
"ResultId": "2037781592655560704"
}
```

Notify Plate Coupon Record Changes (Including Add, Edit, Delete)

Brief Description

- This API is used to notify the software to add, update, or delete plate coupons.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Notice`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	PlateCoupon	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API

Parameter	Location	Type	Required	Description	Notes
» NoticeType	body	integer	Yes	Notification type	Notification type: 0 - Add or Update, 1 - Delete.
» Items	body	[object]	Yes	Notification data collection	Collection of notification data
»» No	body	string	Yes	Record number	Unique number for the current record.
»» CouponSchemeNo	body	string	Yes	Coupon scheme number	Coupon scheme number.
»» Type	body	string	Yes	Coupon discount type	Coupon discount type (101 - Discount amount, 102 - Discount duration, 103 - Discount ratio, 104 - Free until specified time)
»» Value	body	number	Yes	Discount value	For discount amount: the amount in yuan.
»» StartTime	body	string null	Yes	Start time	Start time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» EndTime	body	string null	Yes	End time	End time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» OrderNo	body	string null	Yes	Parking order	Unique parking order created when the vehicle enters.
»» LicensePlateNumber	body	string null	Yes	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default).
»» BusinessName	body	string	Yes	Merchant name	Merchant name.
»» CreationTime	body	string	Yes	Creation time	Creation time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» Remarks	body	string null	Yes	Remarks information	Remarks information.

Request Example

```
1  {
2    "AppId": "xb5glg0hgZ40",
3    "Data": {
4      "NoticeType": 0,
5      "Items": [
6        {
7          "No": "32000020140129785X",
8          "CouponSchemeNo": "test",
9          "Type": "101",
10         "Value": 10,
11         "StartTime": "2026-03-28 14:42:35",
12         "EndTime": "2026-03-28 14:42:35",
13         "OrderNo": "1668010507727935-B12345",
14         "LicensePlateNumber": "B12345",
15         "BusinessName": "test",
16         "CreationTime": "2026-03-28 14:42:35",
17         "Remarks": "test"
18       }
19     ]
20   },
21   "ParkKey": "1772492942362538",
22   "TimeStamp": "1774680172647",
23   "Nonce": "enu5putm0vyf",
24   "Sign": "vn2tmadrbhgu773/kayfr9kabunwb5kqasxbm55vwca="
25 }
```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.

Return Example

```
{
  "ResultCode": 200,
  "ResultMsg": "Add or update operation successfully!",
  "ResultId": "2037782456908677120"
}
```

Notify Coupon Scheme Changes (Including Add, Edit, Delete)

Brief Description

- This API is used to manage the addition, deletion, and modification of coupon schemes.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/Notice`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Query Parameters

Parameter	Type	Example Value	Description
Route	string	CouponScheme	API route

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» NoticeType	body	integer	Yes	Notification type	Notification type: 0 - Add or Update, 1 - Delete.
» Items	body	[object]	Yes	Notification data collection	Collection of notification data
»» No	body	string	Yes	Record number	Unique number for the current record.

Parameter	Location	Type	Required	Description	Notes
»» Name	body	string null	Yes	Coupon scheme name	Coupon scheme name.
»» Type	body	string null	Yes	Coupon discount type	Coupon discount type (101 - Discount amount, 102 - Discount duration, 103 - Discount ratio, 104 - Free until specified time)
»» Amount	body	number null	Yes	Discount amount (for amount type)	Discount amount for amount type (unit: yuan).
»» Duration	body	number	Yes	Discount duration (for duration type)	Discount duration for duration type (unit: minutes).
»» Ratio	body	number	Yes	Discount ratio (for ratio type)	Discount ratio for ratio type.
»» VaildTime	body	string	Yes	Free until specified time	Free until specified time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» Enable	body	integer	Yes	Whether coupon scheme is enabled	Whether coupon scheme is enabled: 0 - Disabled, 1 - Enabled.
»» Scene	body	integer	Yes	Coupon scheme usage scene	Coupon scheme usage scene: 0 - Only for in-field discount, 1 - Only for exit discount, 2 - Exit and in-field discount.
»» CreationTime	body	string	Yes	Creation time	Creation time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
»» OperatorName	body	string null	Yes	Operator name	Name of the operator.

Request Example

```

1  {
2    "AppId": "xb5glg0hg40",
3    "Data": {
4      "NoticeType": 0,
5      "Items": [
6        {
7          "No": "710000197308123736",
8          "Name": "test",

```

```

9      "Type": "101",
10     "Amount": 10,
11     "VaildTime": "2026-03-28 14:45:09",
12     "Enable": 1,
13     "Scene": 1,
14     "CreationTime": "2026-03-28 14:45:09"
15   }
16 ]
17 },
18 "ParkKey": "1772492942362538",
19 "TimeStamp": "1774680383712",
20 "Nonce": "vw68p8xvrpdm",
21 "Sign": "8dicgrq2fyvczw4hygktnoapaycjthehtlemxzozf18="
22 }

```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.

Return Example

```

{
  "ResultCode": 200,
  "ResultMsg": "Add or update operation successfully!",
  "ResultId": "2037783342175256576"
}

```

Notify Control Lane Barrier Status (Including Release, Close,Normally Release,Cancel Normally Release)

Brief Description

- This API is used to notify and control the lane barrier gate status.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/SetLaneGateStatus`

Request Method

- POST

Request Content-Type

- application/json

Request Parameters

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» LaneNo	body	string	Yes	Lane number	Lane number
» Status	body	integer	Yes	Set barrier gate status	Set barrier gate status: 0 - Close gate, 1 - Open gate, 2 - Control constant open (if the gate is already in constant open state, sending this command will cancel constant open).
» OperatorName	body	string null	Yes	Operator name	Name of the operator.

Request Example

```
1  {
2    "AppId": "xb5glg0hg40",
3    "Data": {
4      "LaneNo": "2",
5      "Status": 1
6    },
7    "ParkKey": "1772492942362538",
8    "TimeStamp": "1774680773498",
9    "Nonce": "qx04njd4zmx1",
10   "Sign": "k2nuvwcw91kcrnaiwz+gqyxexo2wt7cfrdfksbxfua="
11 }
```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.

Return Example

```
{
  "ResultCode": 200,
  "ResultMsg": "[Exit Camera]Release gate successfully!",
  "ResultId": "2037784977043652608"
}
```

Notify Lane Popup Order Fee Modification

Brief Description

- This API is used to modify the exit fee for a license plate by a third-party platform when the vehicle owner requests assistance from the platform after vehicle recognition.
- The modified amount must be less than the amount displayed in the lane popup window.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/ModifyLaneOrderMoney`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot

Parameter	Location	Type	Required	Description	Notes
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» OrderNo	body	string null	Yes	Parking order	Unique parking order created when the vehicle enters.
» LaneNo	body	string	Yes	Lane number	Lane number.
» Money	body	integer	Yes	Amount to modify the lane fee to	Amount to modify the lane fee to (unit: cents).

Request Example

```

1  {
2    "AppId": "xb5glg0hgZ40",
3    "Data": {
4      "OrderNo": "1774695602882864-BCDE11",
5      "LaneNo": "2",
6      "Money": 10
7    },
8    "ParkKey": "1772492942362538",
9    "TimeStamp": "1774683226943",
10   "Nonce": "v9hilqn7bkr7",
11   "Sign": "1114iggtxxuowa1ox65h9fsa0fm7be8s6nmdmvdevcc="
12 }

```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.

Return Example

```
{
  "ResultCode": 200,
  "ResultMsg": "Update payment amount successfully!",
  "ResultId": "2037795267470721024"
}
```

Notify Vehicle Payment Information

Brief Description

- This API is used to send payment notification information to the system.
- After receiving the notification, the exit lane will open the gate.
- Note: Payment order numbers cannot be duplicated. If a duplicate order number is received, the system will return a message indicating that the payment order is already being processed.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/PaySuccess`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» OrderNo	body	string	Yes	Parking order	Unique parking order created when the vehicle enters.

Parameter	Location	Type	Required	Description	Notes
» TotalAmountDue	body	number	Yes	Total amount due	Total amount due (in yuan, accurate to two decimal places).
» TotalAmountPaid	body	number	Yes	Total amount paid	Total amount paid (in yuan, accurate to two decimal places).
» DiscountAmount	body	number	Yes	Discount amount	Discount amount (in yuan, accurate to two decimal places).
» PaymentTime	body	string	Yes	Payment time	Payment time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
» PaymentMethod	body	string	Yes	Payment method number	For details, refer to the type enumeration/payment method enumeration information.
» PaymentOrderNo	body	string	Yes	Payment order number	Payment order number.
» Remarks	body	string null	Yes	Remarks information	Remarks information.
» PlateCouponNos	body	[object]	Yes	Collection of used license plate coupons	Collection of used license plate coupons.
»» No	body	string	Yes	Coupon number	Coupon number.

Request Example

```
1  {
2    "AppId": "xb5glg0hgza40",
3    "Data": {
4      "OrderNo": "1774695602882864-BCDE11",
5      "TotalAmountDue": 5,
6      "TotalAmountPaid": 5,
7      "DiscountAmount": 0,
8      "PaymentTime": "2026-03-28 15:57:36",
9      "PaymentMethod": "79024",
10     "PaymentOrderNo": "521774684656270152"
11   },
12   "ParkKey": "1772492942362538",
13   "TimeStamp": "1774684701374",
14   "Nonce": "nhrqcaxymc9c",
15   "Sign": "vaimlh60+n6sf3hghy/ps53z4wgzpjbc/0iwuhahzbo="
16 }
```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.

Return Example

```
{
  "ResultCode": 200,
  "ResultMsg": "Payment successfully!",
  "ResultId": "2037801451640619008"
}
```

Create Entry Order for Vehicle Without Plate

Brief Description

- This API is used to create an entry record for vehicles without a license plate upon entry and return the parking order number.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/CreateNoPlate`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot

Parameter	Location	Type	Required	Description	Notes
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» LicensePlateNumber	body	string	Yes	License plate number	Virtual license plate number for license-plate-less vehicle entry, preferably not a real license plate.
» EntryTime	body	string	Yes	Entry time	Entry time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
» VehicleTypeNo	body	string null	Yes	Vehicle type number	Vehicle type number.
» ParkingCardTypeNo	body	string	Yes	License plate type number	License plate type number.
» LaneNo	body	string	Yes	Entry lane number	Entry lane number.
» OperatorName	body	string null	Yes	Operator name	Name of the operator.
» Remarks	body	string null	Yes	Remarks information	Remarks information.

Request Example

```
1  {
2      "ParkKey": "{{parkkey}}",
3      "AppId": "{{appid}}",
4      "TimeStamp": "{{timestamp}}",
5      "Nonce": "okulfmutapcl",
6      "Sign": "uxhrbhycphdtruhvqlqosicvc",
7      "Data": {
8          "LicensePlateNumber": "A887755",
9          "EntryTime": "2024-02-27 14:06:46",
10         "VehicleTypeNo": "1708773681734708",
11         "ParkingCardTypeNo": "1708773681731663",
12         "LaneNo": "1",
13         "Remarks": "This is a prominent remark"
14     }
15 }
```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.
» OrderNo	string null	true	Parking order	Unique parking order created when the vehicle enters.

Return Example

```
{
  "ResultCode": 200,
  "ResultMsg": "License-plate-less vehicle entry confirmed successfully (gate opened)!",
  "ResultId": "1762358795399364608",
  "Data": {
    "OrderNo": "1762358795516805120-A887755"
  }
}
```

Notify Callback Parameter Changes

Brief Description

- This API is used to modify callback parameters.
- The `Index` field in the `Data` array corresponds to the sequence number in the callback interface enumeration. Please configure the interfaces that require callbacks according to your needs. Sequence numbers not included in the array will default to no callback. Similarly, an empty URL address will also default to no callback. For interface security specifications, please refer to the API authentication documentation.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/SetCallbackParams`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	[object]	Yes	Specific data	Data sent by the API, in string format.
» Index	body	integer	No	Callback interface enumeration sequence number	For details, refer to the type enumeration/callback interface enumeration.
» Url	body	string	No	Callback interface URL	The callback address for this enumeration sequence interface. If not filled, the callback will not be enabled.
» Interval	body	integer	Yes	Callback data processing interval	Callback data processing interval, unit: milliseconds. Must be $\geq 200\text{ms}$ and $\leq 600000\text{ms}$ (10 minutes).

Request Example

```
1  {
2    "AppId": "xb5glg0hgzz40",
3    "Data": [
4      {
5        "Index": 4,
6        "Url": "http://192.168.22.247:8080",
7        "Interval": 1000
8      }
9    ],
10   "ParkKey": "1772492942362538",
11   "TimeStamp": "1774688483419",
12   "Nonce": "50urf7owudi4",
13   "Sign": "ja2bc5hozy6lw5fwldr8e0u/7d5cacmb7nc9hrb0lso="
14 }
```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.

Return Example

```
{
  "ResultCode": 200,
  "ResultMsg": "Operation successfully",
  "ResultId": "2037817314582233088"
}
```

Notify Failed Callback Business Retry

Brief Description

- Calling this API triggers the parking local open API service to re-push callback business that failed within the specified date range.
- This API can be called only once every 30 minutes.

Request URL

- `http://localhost:8888/Api/V2/OpenApi/RetryCallback`

Request Method

- POST

Request Content-Type

- `application/json`

Request Parameters

Body Parameters

Parameter	Location	Type	Required	Description	Notes
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot

Parameter	Location	Type	Required	Description	Notes
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Payload	Data object specific to this API
» StartTime	body	string null	Yes	Start time	Start time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
» EndTime	body	string null	Yes	End time	End time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51).
» RefreshTypes	body	[object]	No	Callback types to retry	Refer to the sequence numbers in the type enumeration/callback interface enumeration. If empty, all callback types will be retried.
»» Index	body	integer	Yes	Callback interface enumeration sequence number	For details, refer to the type enumeration/callback interface enumeration.

Request Example

```
1  {
2    "AppId": "xb5glg0hg40",
3    "Data": {
4      "StartTime": "2026-03-28 16:06:28",
5      "EndTime": "2026-03-28 17:06:28",
6      "RefreshTypes": [
7        {
8          "Index": 1
9        }
10     ]
11   },
12   "ParkKey": "1772492942362538",
13   "TimeStamp": "1774688790638",
14   "Nonce": "6af6hvf93b1",
15   "Sign": "7nyutdsajan+ffrftxibj+kej8rvajgq0zslcxond+4="
16 }
```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.

Return Example

```
{
  "ResultCode": 200,
  "ResultMsg": "Re-pull failed callback business data successfully 0",
  "ResultId": "2037818603147919360"
}
```

6. Callback Interfaces (V2.0&V3.0)

Vehicle Entry Callback

Brief Description

- Specification for the vehicle entry event notification interface.
- If this interface is not implemented, please do not enable it in the callback parameter settings to avoid unnecessary data notification operations.

Example URL

- `{{Third-party URL}}`, [http://localhost:8888/API/V2/OnVehicleIn`](http://localhost:8888/API/V2/OnVehicleIn) 

Note: This URL is the callback URL from the system to the third party, i.e., the endpoint where the third party receives data pushed by the system.

Request Method

- POST

Request Content-Type

- `application/json`

Callback Parameter Description

Parameter	Type	Name	Required	Description	Note
ParkKey	body	string	yes	Parking Lot Key	Local parking lot identifier
Appld	body	string	yes	App Service ID	Local parking app service identifier
TimeStamp	body	string	yes	Timestamp	13-digit timestamp, accurate to milliseconds
Nonce	body	string	yes	Nonce String	Random string
Sign	body	string	yes	API Signature	Signature generated according to API authentication
Data	body	object	yes	Data Detail	Data object corresponding to the API request
» OrderNo	body	string	yes	Parking Order No.	Unique parking order created on vehicle entry
» LicensePlateNumber	body	string	yes	License Plate Number	Converted to uppercase by default
» EntryTime	body	string	yes	Entry Time	Format: yyyy-MM-dd HH:mm:ss
» LicensePlateColor	body	integer	yes	License Plate Color Code	Refer to license plate color enum
» EntryLaneNo	body	string	yes	Entry Lane No.	Entry lane number
» EntryLaneName	body	string	yes	Entry Lane Name	Entry lane name
» EntryOperatorName	body	string	yes	Entry Operator	Entry operator name
» VehicleTypeNo	body	string	yes	Vehicle Type No.	Vehicle type code
» VehicleTypeName	body	string	yes	Vehicle Type Name	Vehicle type name
» ParkingCardTypeNo	body	string	yes	Plate Type No.	Plate type code
» ParkingCardTypeName	body	string	yes	Plate Type Name	Plate type name
» ReserveOrderNo	body	string!null	yes	Reservation Order No.	Available if the vehicle is a reservation vehicle
» ParkingAreaNo	body	string	yes	Parking Area No.	Current parking area code
» ParkingAreaName	body	string	yes	Parking Area Name	Current parking area name

Parameter	Type	Name	Required	Description	Note
» EntryImageUrl	body	string null	yes	Entry Image URL	Cloud storage URL if cloud image storage is enabled
» EntryImageBase64	body	string null	yes	Entry Image Base64	Base64 of entry image; excluded from signature

Request Example

```

1  {
2      "ParkKey": "1772492942362538",
3      "AppId": "xb5glg0hgZ40",
4      "Data": {
5          "OrderNo": "1774718489366642-6Y8MT1",
6          "LicensePlateNumber": "76Y8MT1",
7          "EntryTime": "2026-03-28 17:21:27",
8          "LicensePlateColor": 1,
9          "EntryLaneNo": "1",
10         "EntryLaneName": "Entrance Lane",
11         "EntryOperatorName": "qq",
12         "VehicleTypeNo": "1772492942395203",
13         "VehicleTypeName": "Blue Plate Vehicle",
14         "ParkingCardTypeNo": "1772492942389506",
15         "ParkingCardTypeName": "Temporary Car A",
16         "ParkingAreaNo": "1772492942540835",
17         "ParkingAreaName": "Outer Field",
18         "EntryImageUrl": "",
19         "EntryImageBase64": ""
20     },
21     "Sign": "zngg3e7rcch9dst41oaem9j+pmvulcdpbf1rghegoy=",
22     "TimeStamp": "1774689712754",
23     "Nonce": "2037822471420280832",
24     "RequestId": "2037822471407697920"
25 }

```

Return Parameter Description

Parameter	Type	Required	Description	Remarks
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default.
ResultMsg	string	true	Request message	Information message returned by the API.
ResultId	string	true	Request response identifier	Unique identifier for this request processing.
Data	object null	false	Response data	Data object returned by the API.

Return Example

```
{
  "ResultCode": 200,
  "ResultMsg": "Operation successful",
  "ResultId": 500303426998439
}
```

Vehicle Exit Callback

Brief Description

- Specification for the vehicle exit event notification interface.
- If this interface is not implemented, please do not enable it in the callback parameter settings to avoid unnecessary data notification operations.

Example URL

- `{{Third-party URL}}`, Example URL: `http://localhost:8888/API/V2/OnVehicleOut`

Note: This URL is the callback URL from the system to the third party, i.e., the endpoint where the third party receives data pushed by the system.

Request Method

- POST

Content Type

- Content-Type: `application/json`

Callback Parameter Description

Parameter	Type	Required	Description	Notes	
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Specific data	Data object sent by the API

Parameter	Type	Required	Description	Notes	
» OrderNo	body	string null	Yes	Parking order	Unique parking order created when the vehicle enters
» LicensePlateNumber	body	string null	Yes	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default)
» ExitTime	body	string null	Yes	Exit time	Exit time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51)
» LicensePlateColor	body	integer	Yes	License plate color sequence number	For details, refer to the sequence number in the enumeration type/license plate color enumeration
» ExitLaneNo	body	string null	Yes	Exit lane number	Exit lane number
» ExitLaneName	body	string null	Yes	Exit lane name	Exit lane name
» ExitOperatorName	body	string null	Yes	Exit operator	Exit operator name
» VehicleTypeNo	body	string null	Yes	Vehicle type number	Vehicle type number
» VehicleTypeName	body	string null	Yes	Vehicle type name	Vehicle type name
» ParkingCardTypeNo	body	string null	Yes	License plate type number	License plate type number
» ParkingCardTypeName	body	string null	Yes	License plate type name	License plate type name
» ParkingAreaNo	body	string	Yes	Parking area number	Current parking area number
» ParkingAreaName	body	string null	Yes	Parking area name	Current parking area name
» ExitImageUrl	body	string null	Yes	Exit passage capture image URL	Exit passage capture image URL; if cloud platform is enabled with image storage, this is a cloud-stored image URL

Parameter	Type	Required	Description	Notes	
» ExitImageBase64	body	string null	Yes	Exit passage capture image	Exit passage capture image Base64
» TotalAmountDue	body	number	Yes	Total amount due	Total amount due (in yuan, accurate to two decimal places)
» TotalAmountPaid	body	number	Yes	Total amount paid	Total amount paid (in yuan, accurate to two decimal places)
» IsFreePass	body	boolean	Yes	Whether free pass is granted	Whether free pass is granted
» FreePassReason	body	string null	Yes	Free pass reason	Reason for granting free pass when vehicle is allowed free passage
» Remarks	body	string null	Yes	Remarks information	Remarks information

Callback Parameter Example

```

1  {
2      "ParkKey": "1772492942362538",
3      "AppId": "xb5glg0hg240",
4      "Data": {
5          "OrderNo": "1774718489366642-6Y8MT1",
6          "LicensePlateNumber": "76Y8MT1",
7          "ExitTime": "2026-03-28 17:28:26",
8          "LicensePlateColor": 1,
9          "ExitLaneNo": "2",
10         "ExitLaneName": "Exit Lane",
11         "ExitOperatorName": "qq",
12         "VehicleTypeNo": "1772492942395203",
13         "VehicleTypeName": "Blue Plate Vehicle",
14         "ParkingCardTypeNo": "1772492942389506",
15         "ParkingCardTypeName": "Temporary Car A",
16         "ParkingAreaNo": "1772492942540835",
17         "ParkingAreaName": "Outer Field",
18         "ExitImageUrl": "",
19         "ExitImageBase64": "",
20         "TotalAmountDue": 5,
21         "TotalAmountPaid": 5,
22         "IsFreePass": false
23     },
24     "Sign": "asoekfeybydqgmhxxchhn8jraddnzfcls2+zdtwh5zk=",
25     "TimeStamp": "1774690115822",
26     "Nonce": "2037824162010005504",
27     "RequestId": "2037824161997422592"
28 }

```

Callback Return Description

Parameter	Type	Required	Description	Notes
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default
ResultMsg	string	true	Request message	Information message returned by the API
Data	object null	false	Response data	Data object returned by the API

Return Example

```
1  {  
2    "ResultCode": 200,  
3    "ResultMsg": "Operation successful"  
4  }
```

Payment Result Callback

Brief Description

- Specification for the payment result callback notification interface.
- If this interface is not implemented, please do not enable it in the callback parameter settings to avoid unnecessary data notification operations.

Example URL

- `{{Third-party URL}}`, example: `http://localhost:8888/API/V2/OnPayResult`

Note: This URL is the callback URL from the system to the third party, i.e., the endpoint where the third party receives data pushed by the system.

Request Method

- POST

Content Type

- Content-Type: `application/json`

Callback Parameter Description

Parameter	Type	Required	Description	Notes	
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier

Parameter	Type	Required	Description	Notes	
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Specific data	Data object sent by the API
» OrderNo	body	string null	Yes	Parking order	Unique parking order created when the vehicle enters
» PaymentOrderNo	body	string	Yes	Payment order number	Payment order number
» LicensePlateNumber	body	string null	Yes	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default)
» TotalAmountDue	body	number	Yes	Total amount due	Total amount due (in yuan, accurate to two decimal places)
» TotalAmountPaid	body	number	Yes	Total amount paid	Total amount paid (in yuan, accurate to two decimal places)
» DiscountAmount	body	number	Yes	Discount amount	Discount amount (in yuan, accurate to two decimal places)
» PaymentTime	body	string	Yes	Payment time	Payment time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51)
» PaymentMethod	body	string	Yes	Payment method number	For details, refer to the type enumeration/payment method enumeration information

Parameter	Type	Required	Description	Notes	
» PayScene	body	integer	Yes	Payment scene	For details, refer to the type enumeration/payment scene enumeration information
» VehicleTypeNo	body	string null	Yes	Vehicle type number	Vehicle type number
» VehicleTypeName	body	string null	Yes	Vehicle type name	Vehicle type name
» ParkingCardTypeNo	body	string null	Yes	License plate type number	License plate type number
» ParkingCardTypeName	body	string null	Yes	License plate type name	License plate type name
» Remarks	body	string null	Yes	Remarks information	Remarks information
» ParkingDuration	body	integer	Yes	Parking duration	Parking duration (in minutes)
» StoredValueDeduction	body	number	No	Prepaid vehicle deduction amount	Prepaid vehicle deduction amount (in yuan, accurate to two decimal places)
» OperatorName	body	string null	Yes	Operator name	Name of the operator
» CouponUsageList	body	[object]	No	Coupon usage list	List containing the collection of coupon codes used for the current order
»» No	body	string	Yes	Used coupon number	Coupon number that was used

Callback Parameter Example

```

1  {
2      "ParkKey": "1772492942362538",
3      "AppId": "xb5glg0hgZ40",
4      "Data": {
5          "OrderNo": "1774718489366642-6Y8MT1",
6          "PaymentOrderNo": "P01774718907646185",
7          "LicensePlateNumber": "76Y8MT1",
8          "TotalAmountDue": 5,
9          "TotalAmountPaid": 5,
10         "DiscountAmount": 0,
11         "PaymentTime": "2026-03-28 17:28:27",
12         "PaymentMethod": "79001",
13         "PayScene": 0,
14         "VehicleTypeNo": "1772492942395203",

```

```

15         "VehicleTypeName": "Blue Plate Vehicle",
16         "ParkingCardTypeNo": "1772492942389506",
17         "ParkingCardTypeName": "Temporary Car A",
18         "ParkingDuration": 6,
19         "StoredValueDeduction": 0,
20         "OperatorName": "qq"
21     },
22     "Sign": "z3nbuv83rzxsjrmd0+6lkz97ntrrpjl7rgpnh+0ltco=",
23     "TimeStamp": "1774690112810",
24     "Nonce": "2037824149376761856",
25     "RequestId": "2037824149339013120"
26 }

```

Callback Return Description

Parameter	Type	Required	Description	Notes
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default
ResultMsg	string	true	Request message	Information message returned by the API
Data	object null	false	Response data	Data object returned by the API

Return Example

```

1  {
2      "ResultCode": 200,
3      "ResultMsg": "Operation successful"
4  }

```

Vehicle Issuance Callback

Brief Description

- Specification for the vehicle subscription notification interface.
- If this interface is not implemented, please do not enable it in the callback parameter settings to avoid unnecessary data notification operations.

Example URL

- `{{Third-party URL}}`, Example URL: `http://localhost:8888/API/V2/OnSubscribedVehicle`

Note: This URL is the callback URL from the system to the third party, i.e., the endpoint where the third party receives data pushed by the system.

Request Method

- POST

Content Type

- Content-Type: `application/json`

Callback Parameter Description

Parameter	Type	Required	Description	Notes	
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Specific data	Data object sent by the API
» LicensePlateNumber	body	string null	Yes	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default)
» OwnerNo	body	string null	Yes	Owner number	Unique identifier number of the vehicle owner
» OwnerName	body	string null	Yes	Owner name	Name of the vehicle owner recorded in the issued vehicle record
» OwnerParkingSpaceCount	body	integer	Yes	Owner's parking space count	Number of parking spaces owned by the vehicle owner
» StartTime	body	string null	Yes	Start time	Start time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51)
» EndTime	body	string null	Yes	End time	End time, following the date format in basic rules, accurate to

Parameter	Type	Required	Description	Notes	
					seconds (e.g., 2023-08-01 09:34:51)
» StoredVehicleBalance	body	number	Yes	Prepaid vehicle balance	Prepaid vehicle balance (in yuan, accurate to two decimal places)
» VehicleTypeNo	body	string null	Yes	Vehicle type number	Vehicle type number
» VehicleTypeName	body	string null	Yes	Vehicle type name	Vehicle type name
» ParkingCardTypeNo	body	string null	Yes	License plate type number	License plate type number
» ParkingCardTypeName	body	string null	Yes	License plate type name	License plate type name
» CreationTime	body	string	Yes	Creation time	Creation time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51)
» ResidentialAddress	body	string	Yes	Residential address	Residential address of the vehicle owner
» PhoneNumber	body	string	Yes	Phone number	Phone number
» Remarks	body	string null	Yes	Remarks information	Remarks information
» OperatorName	body	string null	Yes	Operator name	Name of the operator

Callback Parameter Example

```
1  {
2    "ParkKey": "1772492942362538",
3    "AppId": "xb5glg0hgza40",
4    "Data": {
5      "LicensePlateNumber": "Q1234",
6      "PaymentOrderNo": "MC1774721023079330",
7      "TotalAmountDue": 0,
8      "TotalAmountPaid": 0,
9      "DiscountAmount": 0,
10     "PaymentMethod": "79001",
11     "PaymentTime": "2026-03-28 18:03:43",
12     "OperatorName": "qq",
13     "StartTime": "2026-03-28 00:00:00",
14     "EndTime": "2026-04-27 23:59:59",
15     "StoredVehicleBalance": 0,
```

```

16         "VehicleTypeNo": "1772492942395203",
17         "VehicleTypeName": "Blue Plate Vehicle",
18         "ParkingCardTypeNo": "1772492942389944",
19         "ParkingCardTypeName": "Monthly Rental Car A"
20     },
21     "Sign": "f0nbrmh+bdhm5y2g0odk6byqhayzpvvet2cwvh0q0gfg=",
22     "TimeStamp": "1774692234475",
23     "Nonce": "2037833048284758016",
24     "RequestId": "2037833048217649152"
25 }

```

Callback Return Description

Parameter	Type	Required	Description	Notes
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default
ResultMsg	string	true	Request message	Information message returned by the API
Data	object null	false	Response data	Data object returned by the API

Return Example

```

1  {
2    "ResultCode": 200,
3    "ResultMsg": "Operation successful"
4  }

```

Vehicle Cancellation Callback

Brief Description

- Specification for the vehicle unsubscription notification interface.
- If this interface is not implemented, please do not enable it in the callback parameter settings to avoid unnecessary data notification operations.

Example URL

- {{Third-party URL}}, Example URL: `http://localhost:8888/API/V2/OnUnsubscribedVehicle`

Note: This URL is the callback URL from the system to the third party, i.e., the endpoint where the third party receives data pushed by the system.

Request Method

- POST

Content Type

- Content-Type: `application/json`

Callback Parameter Description

Parameter	Type	Required	Description	Notes	
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Specific data	Data object sent by the API
» LicensePlateNumber	body	string null	Yes	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default)
» OperatorName	body	string null	Yes	Operator name	Name of the operator
» OperationTime	body	string	Yes	Operation time	Operation time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51)

Callback Parameter Example

```
1  {
2    "ParkKey": "1772492942362538",
3    "AppId": "xb5glg0hg240",
4    "Data": {
5      "LicensePlateNumber": "Q1234",
6      "OperatorName": "",
7      "OperationTime": "2026-03-28 18:07:11"
8    },
9    "Sign": "o8uggl5yd0tswjdnq1ezp1jx+bwhlh5ztlulqtdvis8=",
10   "TimeStamp": "1774692458704",
11   "Nonce": "2037833988769349633",
12   "RequestId": "2037833988769349632"
13 }
```

Callback Return Description

Parameter	Type	Required	Description	Notes
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default
ResultMsg	string	true	Request message	Information message returned by the API
Data	object null	false	Response data	Data object returned by the API

Return Example

```
1  {  
2    "ResultCode": 200,  
3    "ResultMsg": "Operation successful"  
4  }
```

Vehicle Recharge and Extension Callback

Brief Description

- Specification for the vehicle recharge and extension notification interface.
- If this interface is not implemented, please do not enable it in the callback parameter settings to avoid unnecessary data notification operations.

Example URL

- `{{Third-party URL}}`, Example URL: `http://localhost:8888/API/V2/OnRechargeGracePeriod`

Note: This URL is the callback URL from the system to the third party, i.e., the endpoint where the third party receives data pushed by the system.

Request Method

- POST

Content Type

- Content-Type: `application/json`

Callback Parameter Description

Parameter	Type	Required	Description	Notes	
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier

Parameter	Type	Required	Description	Notes	
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Specific data	Data object sent by the API
» LicensePlateNumber	body	string null	Yes	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default)
» PaymentOrderNo	body	string	Yes	Payment order number	Payment order number
» TotalAmountDue	body	number	Yes	Total amount due	Total amount due (in yuan, accurate to two decimal places)
» TotalAmountPaid	body	number	Yes	Total amount paid	Total amount paid (in yuan, accurate to two decimal places)
» DiscountAmount	body	number	Yes	Discount amount	Discount amount (in yuan, accurate to two decimal places)
» PaymentMethod	body	string	Yes	Payment method number	For details, refer to the type enumeration/payment method enumeration information
» PaymentTime	body	string	Yes	Payment time	Payment time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51)
» OperatorName	body	string null	Yes	Operator name	Name of the operator
» StartTime	body	string null	Yes	Start time	Start time, following the date format in basic rules,

Parameter	Type	Required	Description	Notes	
					accurate to seconds (e.g., 2023-08-01 09:34:51)
» EndTime	body	string null	Yes	End time	End time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51)
» StoredVehicleBalance	body	number	Yes	Prepaid vehicle balance	Prepaid vehicle balance (in yuan, accurate to two decimal places)
» VehicleTypeNo	body	string null	Yes	Vehicle type number	Vehicle type number
» VehicleTypeName	body	string null	Yes	Vehicle type name	Vehicle type name
» ParkingCardTypeNo	body	string null	Yes	License plate type number	License plate type number
» ParkingCardTypeName	body	string null	Yes	License plate type name	License plate type name

Callback Parameter Example

```
1  {
2    "ParkKey": "1772492942362538",
3    "AppId": "xb5glg0hgZ40",
4    "Data": {
5      "LicensePlateNumber": "P12345",
6      "PaymentOrderNo": "MC17747215714999401772492942362538",
7      "TotalAmountDue": 20,
8      "TotalAmountPaid": 20,
9      "DiscountAmount": 0,
10     "PaymentMethod": "79001",
11     "PaymentTime": "2026-03-28 18:12:51",
12     "OperatorName": "qq",
13     "StartTime": "2026-12-28 00:00:00",
14     "EndTime": "2027-05-01 23:59:59",
15     "StoredVehicleBalance": 0,
16     "VehicleTypeNo": "1772492942395203",
17     "VehicleTypeName": "Blue Plate Vehicle",
18     "ParkingCardTypeNo": "1772492942389944",
19     "ParkingCardTypeName": "Monthly Rental Car A"
20   },
21   "Sign": "3oqfkqasdhj9jnypuh+etwdf10j5efvqnsml96gzmdi=",
22   "TimeStamp": "1774692776940",
23   "Nonce": "2037835323547877376",
24   "RequestId": "2037835323468185600"
25 }
```

Callback Return Description

Parameter	Type	Required	Description	Notes
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default
ResultMsg	string	true	Request message	Information message returned by the API
Data	object null	false	Response data	Data object returned by the API

Return Example

```
1  {  
2    "ResultCode": 200,  
3    "ResultMsg": "Operation successful"  
4  }
```

Gate Release Record Callback

Brief Description

- Specification for the gate release record notification interface.
- If this interface is not implemented, please do not enable it in the callback parameter settings to avoid unnecessary data notification operations.

Example URL

- `{{Third-party URL}}`, Example URL: `http://localhost:8888/API/V2/OnGateReleaseRecord`

Note: This URL is the callback URL from the system to the third party, i.e., the endpoint where the third party receives data pushed by the system.

Request Method

- POST

Content Type

- Content-Type: `application/json`

Callback Parameter Description

Parameter	Type	Required	Description	Notes	
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier

Parameter	Type	Required	Description	Notes	
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Specific data	Data object sent by the API
» No	body	string	Yes	Record number	Unique number for the current record
» OrderNo	body	string null	Yes	Parking order	For abnormal passage, if a vehicle exits automatically without entry record, a parking order will be created. This represents the created parking order number.
» LicensePlateNumber	body	string null	Yes	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default)
» PassageTime	body	string	Yes	Passage time	Passage time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51)
» Remarks	body	string null	Yes	Remarks information	Remarks information
» ImageBase64	body	string	Yes	Image Base64	Base64 data of the captured image during passage. This field does not participate in signature generation.
» InOutType	body	integer	Yes	Lane entry/exit type	Lane entry/exit type: 0 - Exit, 1 - Entry
» PassageLaneNo	body	string	Yes	Passage lane number	Passage lane number
» PassageLaneName	body	string	Yes	Passage lane name	Passage lane name

Parameter	Type	Required	Description	Notes	
» OperatorName	body	string null	Yes	Operator name	Name of the operator
» AbnormalPassType	body	integer	Yes	Abnormal passage type	Abnormal passage type: 6101 - Automatic release without entry record, 6201 - Manual gate opening, 6202 - Manual gate closing, 6211 - Gate constant open, 6212 - Cancel gate constant open

Callback Parameter Example

```
1  {
2    "ParkKey": "1708773681705596",
3    "AppId": "p226uwb3",
4    "TimeStamp": "1709018037090",
5    "Data": {
6      "No": "1709046823138325",
7      "OrderNo": null,
8      "LicensePlateNumber": null,
9      "PassageTime": "2024-02-27 15:13:39",
10     "Remarks": "[Exit Camera] Gate opened successfully!",
11     "ImageBase64": null,
12     "InOutType": 0,
13     "PassageLaneNo": "2",
14     "PassageLaneName": "Exit Lane",
15     "OperatorName": "admin",
16     "AbnormalPassType": 6200
17   },
18   "Nonce": "1762375499496062976",
19   "Sign": "glbo72hyooartypvwcdct8obxkrxayjlkxvxcmtj2/8="
20 }
```

Callback Return Description

Parameter	Type	Required	Description	Notes
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default
ResultMsg	string	true	Request message	Information message returned by the API
Data	object null	false	Response data	Data object returned by the API

Return Example

```
1  {
2    "ResultCode": 200,
```

```
3   "ResultMsg": "Operation successful"
4   }
```

Parking Space Change Callback

Brief Description

- Specification for the parking space change notification interface.
- If this interface is not implemented, please do not enable it in the callback parameter settings to avoid unnecessary data notification operations.

Example URL

- `{{Third-party URL}}`, Example URL: `http://localhost:8888/API/V2/OnParkingSpaces`

Note: This URL is the callback URL from the system to the third party, i.e., the endpoint where the third party receives data pushed by the system.

Request Method

- POST

Content Type

- Content-Type: `application/json`

Callback Parameter Description

Parameter	Type	Required	Description	Notes	
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Specific data	Data object sent by the API
» TotalSpaces	body	integer	Yes	Total parking spaces in the lot	Total number of parking spaces in the lot

Parameter	Type	Required	Description	Notes	
» RemainingSpaces	body	integer	Yes	Remaining parking spaces in the lot	Number of remaining parking spaces in the lot
» AreaSpaces	body	[object]	Yes	Area parking space information	Area parking space information
»» AreaNo	body	string	Yes	Area number	Area number
»» AreaName	body	string	Yes	Area name	Area name
»» TotalSpaces	body	integer	Yes	Total number of parking spaces	Total number of parking spaces
»» RemainingSpaces	body	integer	Yes	Number of remaining parking spaces	Number of remaining parking spaces
» CarCardTypeSpaces	body	[object]	Yes	License plate type parking space count	License plate type parking space count
»» AreaCarCardTypeName	body	string	Yes	Area name + License plate type name	Area name + License plate type name, e.g., Outer Field Temporary Vehicle A
»» AreaNo	body	string	Yes	Area number	Area number
»» UseSpaces	body	integer	Yes	Number of occupied parking spaces	Number of occupied parking spaces, i.e., the number of vehicles of this license plate type parked in the area

Callback Parameter Example

```

1  {
2      "ParkKey": "1772492942362538",
3      "AppId": "xb5glg0hg240",
4      "Data": {
5          "TotalSpaces": 500,
6          "RemainingSpaces": 499,
7          "AreaSpaces": [
8              {
9                  "AreaNo": "1772492942540835",
10                 "AreaName": "Outer Field",
11                 "TotalSpaces": 500,
12                 "RemainingSpaces": 499
13             }
14         ],
15         "CarCardTypeSpaces": [

```

```

16         {
17             "AreaCarCardTypeName": "[Outer Field]Temporary Car A",
18             "AreaNo": "1772492942540835",
19             "UseSpaces": 1
20         }
21     ]
22 },
23 "Sign": "w5msyfgltx69gtf5ovtfkeflvvgh+txai+1j6qitxw0=",
24 "TimeStamp": "1774690132438",
25 "Nonce": "2037824231702560768",
26 "RequestId": "2037824231698366464"
27 }

```

Callback Return Description

Parameter	Type	Required	Description	Notes
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default
ResultMsg	string	true	Request message	Information message returned by the API
Data	object null	false	Response data	Data object returned by the API

Return Example

```

1  {
2      "ResultCode": 200,
3      "ResultMsg": "Operation successful"
4  }

```

Blacklist Callback

Brief Description

- Specification for the blacklist change notification interface.
- If this interface is not implemented, please do not enable it in the callback parameter settings to avoid unnecessary data notification operations.

Example URL

- {{Third-party URL}}, Example URL: `http://localhost:8888/API/V2/OnBlackList`

Note: This URL is the callback URL from the system to the third party, i.e., the endpoint where the third party receives data pushed by the system.

Request Method

- POST

Content Type

- Content-Type: `application/json`

Callback Parameter Description

Parameter	Type	Required	Description	Notes	
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Specific data	Data object sent by the API
» LicensePlateNumber	body	string null	Yes	License plate number	License plate number (Note: when input, all letters are converted to uppercase by default)
» OwnerName	body	string null	Yes	Owner name	Name of the vehicle owner recorded in the blacklist record
» PhoneNumber	body	string	Yes	Phone number	Phone number
» StartTime	body	string null	Yes	Start time	Start time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51)
» EndTime	body	string null	Yes	End time	End time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51)
» CreationTime	body	string	Yes	Creation time	Creation time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51)

Parameter	Type	Required	Description	Notes	
» OperatorName	body	string null	Yes	Operator name	Name of the operator
» Remarks	body	string null	Yes	Remarks information	Remarks information
» Status	body	integer	Yes	Blacklist information	Blacklist information

Callback Parameter Example

```
1  {
2    "ParkKey": "1772492942362538",
3    "AppId": "xb5glg0hgzz40",
4    "Data": {
5      "LicensePlateNumber": "B12345",
6      "OwnerName": "",
7      "PhoneNumber": "",
8      "StartTime": "2026-03-28 14:39:10",
9      "EndTime": "2026-03-28 14:39:10",
10     "CreationTime": "2026-03-28 14:39:10",
11     "Remarks": "",
12     "OperatorName": "OpenAPI Platform-",
13     "Status": 0
14   },
15   "Sign": "zgq0sdnnxahbtfkfa2v+wd1ycmzafx2ywwrztn7juiq=",
16   "TimeStamp": "1774693473977",
17   "Nonce": "2037838247132954624",
18   "RequestId": "2037838247120371712"
19 }
```

Callback Return Description

Parameter	Type	Required	Description	Notes
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default
ResultMsg	string	true	Request message	Information message returned by the API
Data	object null	false	Response data	Data object returned by the API

Return Example

```
1  {
2    "ResultCode": 200,
3    "ResultMsg": "Operation successful"
4  }
```

Booth Lane Information Callback

Brief Description

- Specification for the booth lane information change notification interface.
- If this interface is not implemented, please do not enable it in the callback parameter settings to avoid unnecessary data notification operations.
- When the lane information of a booth is updated, the complete lane information will be sent to the third-party parking service.

Example URL

- `{{Third-party URL}}`, Example URL: `http://localhost:8888/API/V2/OnLaneChange`

Note: This URL is the callback URL from the system to the third party, i.e., the endpoint where the third party receives data pushed by the system.

Request Method

- POST

Content Type

- Content-Type: `application/json`

Callback Parameter Description

Parameter	Type	Required	Description	Notes	
ParkKey	body	string	Yes	Parking Lot ID	Local parking lot identifier
Appld	body	string	Yes	Application Service Identifier	Local application service identifier for the parking lot
TimeStamp	body	string	Yes	Timestamp	13-digit timestamp in milliseconds
Nonce	body	string	Yes	Random String	Randomly generated string
Sign	body	string	Yes	API Signature	Signature generated according to API authentication rules
Data	body	object	Yes	Specific data	Data object sent by the API
» SentryHostNo	body	string	Yes	Booth number	Booth number
» Lanes	body	[object]	Yes	Booth lane information	Booth lane information

Parameter	Type	Required	Description	Notes	
»» Uid	body	integer	Yes	Record primary key	Unique identifier for the current record
»» No	body	string	Yes	Record number	Unique number for the current record
»» LaneName	body	string null	Yes	Lane name	Lane name
»» LaneType	body	integer	Yes	Lane type	Lane type: 1 - Vehicle lane, 2 - Non-motor vehicle lane (Pedestrian lane)
»» LaneDutyMode	body	integer	Yes	Lane duty type	Lane duty type: 0 - Unmanned lane, 1 - On-site duty
»» CameraMode	body	integer	Yes	Camera mode	Camera mode: 1 - Single camera mode, 2 - Dual camera mode, 3 - Multi-camera mode
»» IdInterval	body	integer	Yes	Camera recognition interval	Recognition interval (seconds) (Effective for [Dual camera mode, Multi-camera mode])
»» IsBackCar	body	integer null	Yes	Whether reverse vehicle detection is enabled	Whether reverse vehicle detection is enabled: 0 - Disabled, 1 - Enabled
»» IsCharge	body	integer	Yes	Lane fee collection enabled	Whether lane fee collection is enabled: 0 - Disabled, 1 - Enabled
»» CreationTime	body	string	Yes	Creation time	Creation time, following the date format in basic rules, accurate to seconds (e.g., 2023-08-01 09:34:51)
»» IsSense	body	integer	Yes	Whether ground loop detection is enabled for license-plate-less vehicle scanning	Whether ground loop detection is enabled for license-plate-less vehicle scanning: 0 - Disabled, 1 - Enabled
»» IsSameInOut	body	integer	Yes	Whether same entry/exit mode	Whether same entry/exit mode: 0 - No, 1 - Yes

Parameter	Type	Required	Description	Notes	
»» SamelnOutLaneNo	body	string null	Yes	Associated lane number in same entry/exit mode	Associated lane number in same entry/exit mode
»» SamelnOutType	body	integer	Yes	Entry/exit type of lane in same entry/exit mode	Entry/exit type of lane in same entry/exit mode: 0 - Entry type, 1 - Exit type
»» IsEnableBoard	body	integer	Yes	Whether lane control board is enabled	Whether lane control board is enabled: 0 - Disabled, 1 - Enabled
»» LaneLinkAreas	body	[object]	Yes	Area information bound to the current lane	Area information bound to the current lane
»»» No	body	string	Yes	Record number	Unique number for the current record
»»» AreaNo	body	string	Yes	Associated area number	Associated area number
»»» InOutMode	body	integer	Yes	Entry/exit mode of lane in the current area	Entry/exit mode of lane in the current area: 0 - Exit lane, 1 - Entry lane

Callback Parameter Example

```

1  {
2    "ParkKey": "1772492942362538",
3    "AppId": "xb5glg0hgZ40",
4    "Data": {
5      "SentryHostNo": "1772492942363478",
6      "Lanes": [
7        {
8          "Uid": 1,
9          "No": "1",
10         "LaneName": "Entrance Lane",
11         "LaneType": 1,
12         "IdInterval": 3,
13         "IsBackCar": 0,
14         "IsCharge": 0,
15         "OpenGateAndVoice": 1,
16         "CreationTime": "2026-03-02 23:09:02",
17         "IsSense": 1,
18         "IsSameInOut": 0,
19         "SameInOutType": 0,
20         "IsEnableBoard": 0,
21         "LaneLinkAreas": [
22           {
23             "No": "1774609216714465",
24             "AreaNo": "1772492942540835",
25             "InOutMode": 1
26           }
27         ]
28       }
29     }
30   }

```

```

27         ]
28     },
29     {
30         "Uid": 2,
31         "No": "2",
32         "LaneName": "Exit Lane",
33         "LaneType": 1,
34         "IdInterval": 3,
35         "IsBackCar": 0,
36         "IsCharge": 1,
37         "OpenGateAndVoice": 1,
38         "CreationTime": "2026-03-02 23:09:02",
39         "IsSense": 1,
40         "IsSameInOut": 0,
41         "SameInOutType": 0,
42         "IsEnableBoard": 0,
43         "LaneLinkAreas": [
44             {
45                 "No": "1774609216715573",
46                 "AreaNo": "1772492942540835",
47                 "InOutMode": 0
48             }
49         ]
50     }
51 ]
52 },
53 "Sign": "vuh/cxlb1mjf+hy+xits4ugm0g8zjdpokkz7wj5ihry=",
54 "TimeStamp": "1774693148730",
55 "Nonce": "2037836882948161536",
56 "RequestId": "2037836882931384320"
57 }

```

Callback Return Description

Parameter	Type	Required	Description	Notes
ResultCode	integer	true	Request result code	For details, refer to the result code enumeration. 200 indicates success by default
ResultMsg	string	true	Request message	Information message returned by the API
Data	object null	false	Response data	Data object returned by the API

Return Example

```

1  {
2    "ResultCode": 200,
3    "ResultMsg": "Operation successful"
4  }

```

7. V3.0 Interface Documentation (MQTT)

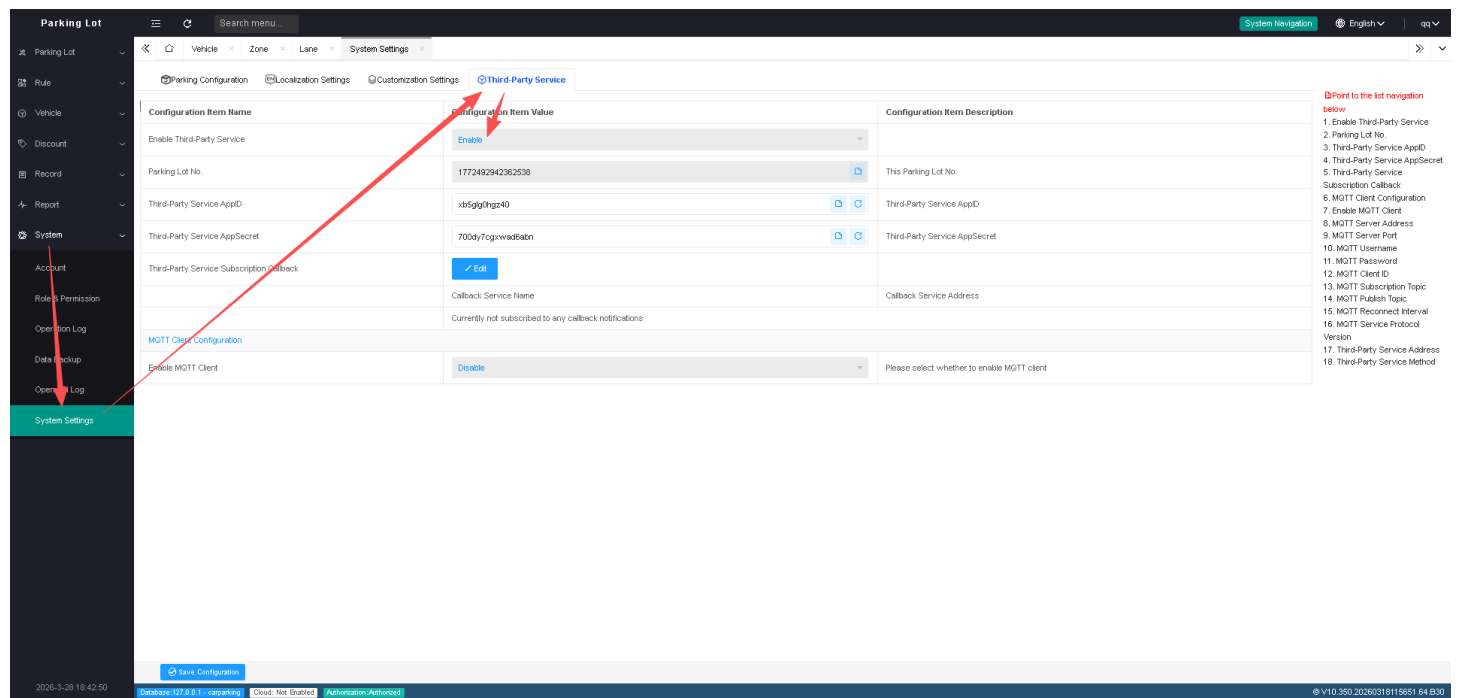
All 3.0 MQTT interfaces follow the same request parameters, response structure, signature rules, and service fields as the 2.0 HTTP interface. Only the transmission method, routing identifier, and additional fields differ, as shown below.

7.1 Development Guide

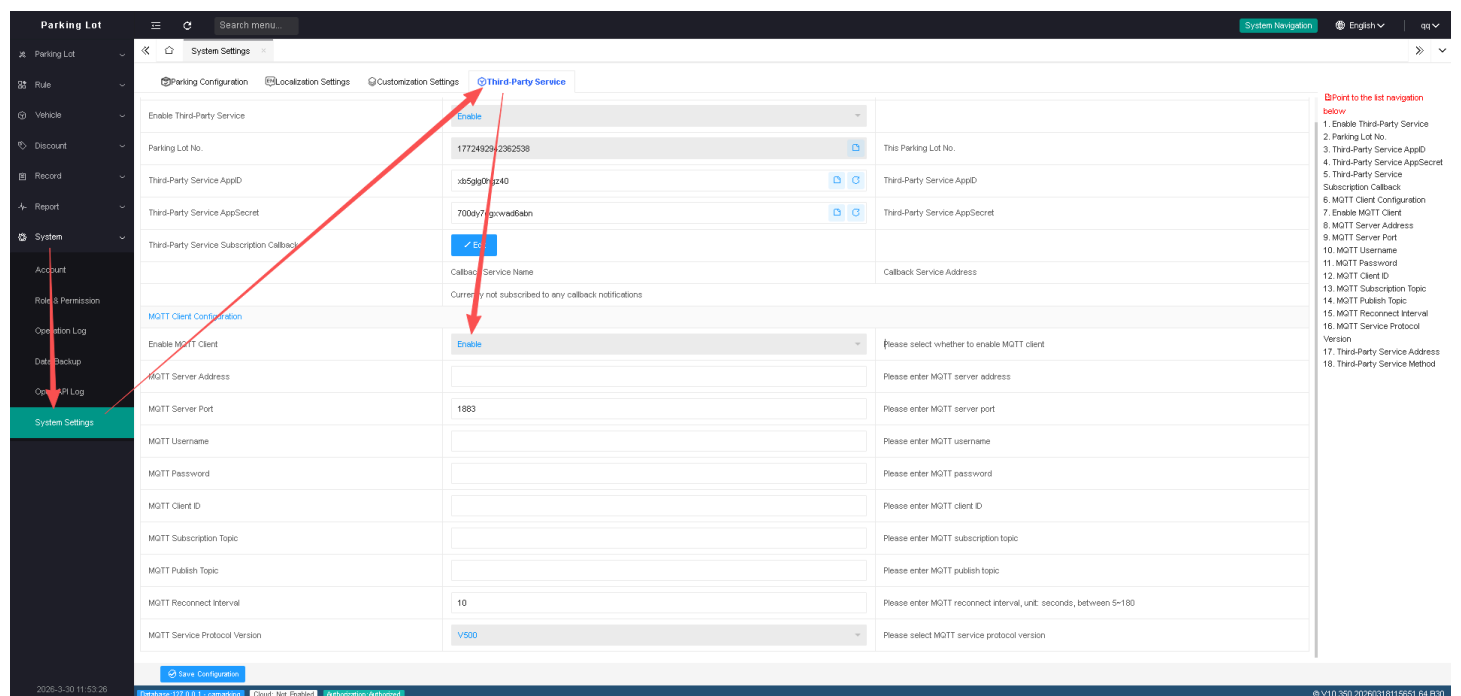
- Please refer to Section 3.1 Development Guide of this document.

7.1.1 Parameter Acquisition and Configuration

Calling the API requires the use of the `appid`, `AppSecret` and `ParkKey` parameters. These must be configured and filled in within the System Settings. The specific steps are shown below:



Enable and configure MQTT parameters



7.1.2 Basic Rules

- Please refer to Section 3.1.2 Development Guide of this document.

7.1.3 Interface Authentication

Note: The MQTT interface uses the same signature as V2; it only adds `Route` and `RequestId` to the request parameters.

- Take the **Query Zone Information** interface as an example.

Note: The following example is only for demonstrating the signature calculation process. In actual use, replace it with real parameters.

1. Assume the request data is as follows:

```
1  {
2      "ParkKey": "{{parkkey}}",
3      "AppId": "{{appid}}",
4      "TimeStamp": "{{timestamp}}",
5      "Nonce": "owekhjgzkosal",
6      "Sign": "pezgzflbnbrwwcdadjajvphz",
7      "Route": "AreaQuery",
8      "RequestId": "12345665",
9      "Data": {
10         "PageIndex": 1,
11         "PageSize": 25,
12         "OrderBy": "Uid",
13         "OrderType": 0,
14         "QueryCondition": [
15             ]
16     }
17 }
```

2. Sort the content inside `Data` in **ASCII ascending order**, then convert it to a **lowercase string**:

```
orderby=uid&ordertype=0&pageindex=1&pagesize=25&querycondition=
```

3. Encrypt the sorted data using **MD5**.The MD5 encrypted result of the `Data` field is:

```
0a78297be1cf12b1faf9efa8b74f1c5e
```

4. Concatenate the prepared parameters into a string:

`AppId` , `ParkKey` , `TimeStamp` , `Nonce` , and `Data` .

```
appid=fvytu5eq&data=0a78297be1cf12b1faf9efa8b74f1c5e&parkkey=1726313394251322&timestamp=1728559158205&nonce=836867
```

5. Convert the string from Step 4 to **lowercase**, then sign it using the **HMAC-SHA256** algorithm with the third-party service `AppSecret` as the secret key. The result is the required signature.

0pv/bitn6cuvogmlg8xtgniuytiealcy5fkxtgy7wci=

6. Add `RequestId` and `Route` to the request body, then send the interface request.

```
1  {
2      "ParkKey": "1726313394251322",
3      "AppId": "fvytu5eq",
4      "TimeStamp": "1728559158205",
5      "Nonce": "836867",
6      "Sign": "0pv/bitn6cuvogmlg8xtgniuytiealcy5fkxtgy7wci=",
7      "Route": "AreaQuery",
8      "RequestId": "12345665",
9      "Data": {
10         "PageIndex": 1,
11         "PageSize": 25,
12         "OrderBy": "Uid",
13         "OrderType": 0,
14         "QueryCondition": []
15     }
16 }
```

Response data:

```
1  {
2      "ResultCode": 200,
3      "ResultMsg": "Query succeeded!",
4      "ResultId": null,
5      "Data": {
6          "PageIndex": 1,
7          "PageSize": 25,
8          "TotalCount": 2,
9          "Results": [
10             {
11                 "Uid": 1,
12                 "No": "1726313394367527",
13                 "AreaName": "Outer Field",
14                 "AreaType": 0,
15                 "AreaSpaceNum": 63,
16                 "PUid": 0,
17                 "Level": 0,
18                 "CreationTime": "2024-10-07 14:16:36"
19             },
20             {
21                 "Uid": 2,
22                 "No": "1728281625923837",
23                 "AreaName": "Inner Field",
24                 "AreaType": 1,
25                 "AreaSpaceNum": 500,
26                 "PUid": 1,
27                 "Level": 1,
28                 "CreationTime": "2024-10-07 14:16:36"
29             }
30         ]
31     }
32 }
```

7.1.4 Type Enumeration List

- Please refer to Section 3.1.4 Development Guide of this document.

7.1.5 Sensitive Information Decryption

- Please refer to Section 3.1.5 Development Guide of this document.

7.1.6 Callback Configuration Instructions

- Please refer to Section 3.1.6 Development Guide of this document.

7.2 Interface Route & Difference Table

Note: All interfaces follow the same request parameters, response structure, and signature rules. The field difference only applies to the corresponding interface version.

Interface Type	Interface Name	v2.0 Route Value	v3.0 Route Value	Field Difference (v3.0 vs v2.0)
Query	Query Zone Information	Area	AreaQuery	v2.0: Route in Query Parameters v3.0: Route + RequestId in Body Parameters
Query	Query Booth Information	SentryHost	BoothQuery	v2.0: Route in Query Parameters v3.0: Route + RequestId in Body Parameters
Query	Query Lane Information	Lane	LaneQuery	v2.0: Route in Query Parameters v3.0: Route + RequestId in Body Parameters
Query	Query Plate Type Information	CardType	PlateTypeQuery	v2.0: Route in Query Parameters v3.0: Route + RequestId

Interface Type	Interface Name	v2.0 Route Value	v3.0 Route Value	Field Difference (v3.0 vs v2.0)
				in Body Parameters
Query	Query Plate Color Information	CarType	PlateColorQuery	v2.0: Route in Query Parameters v3.0: Route + RequestId in Body Parameters
Query	Query Lane Ground Loop Barrier Status	--	BarrierStatusQuery	v2.0: Route is none v3.0: Route + RequestId in Body Parameters
Query	Query Lane Current Order	--	LaneOrderQuery	v2.0: Route is none v3.0: Route + RequestId in Body Parameters
Query	Query Vehicle Record	MonthlyCar	VehicleRecordQuery	v2.0: Route in Query Parameters v3.0: Route + RequestId in Body Parameters
Query	Query Blacklist Records	BlackList	BlacklistQuery	v2.0: Route in Query Parameters v3.0: Route + RequestId in Body Parameters
Query	Query Parking Order Records	ParkingOrder	ParkingOrderQuery	v2.0: Route in Query Parameters v3.0: Route + RequestId

Interface Type	Interface Name	v2.0 Route Value	v3.0 Route Value	Field Difference (v3.0 vs v2.0)
				in Body Parameters
Query	Query Payment Records	PaymentOrder	PaymentRecordQuery	v2.0: Route in Query Parameters v3.0: Route + RequestId in Body Parameters
Query	Query Payment Detail Records	PaymentOrderDetail	PaymentDetailQuery	v2.0: Route in Query Parameters v3.0: Route + RequestId in Body Parameters
Query	Query Gate Release Record	OpenGate	GateReleaseQuery	v2.0: Route in Query Parameters v3.0: Route + RequestId in Body Parameters
Query	Query Coupon Scheme	CouponScheme	CouponSchemeQuery	v2.0: Route in Query Parameters v3.0: Route + RequestId in Body Parameters
Query	Query Plate Coupon Records	PlateCoupon	PlateCouponQuery	v2.0: Route in Query Parameters v3.0: Route + RequestId in Body Parameters
Query	Query Parking Fee for Specified Order	--	OrderFeeQuery	v2.0: Route is none v3.0: Route + RequestId

Interface Type	Interface Name	v2.0 Route Value	v3.0 Route Value	Field Difference (v3.0 vs v2.0)
				in Body Parameters
Notification	In-Parking Record Change Notification (Including Modify and Close)	ModifyParkingOrder	ParkingRecordNotify	v2.0: Route in Query Parameters v3.0: Route + RequestId in Body Parameters
Notification	Notify Vehicle Changes (Including Add, Edit, Delete)	MonthlyCar	VehicleChangeNotify	v2.0: Route in Query Parameters v3.0: Route + RequestId in Body Parameters
Notification	Notify Blacklist Changes (Including Add, Edit, Delete)	BlackList	BlacklistChangeNotify	v2.0: Route in Query Parameters v3.0: Route + RequestId in Body Parameters
Notification	Notify Plate Coupon Record Changes (Including Add, Edit, Delete)	PlateCoupon	PlateCouponChangeNotify	v2.0: Route in Query Parameters v3.0: Route + RequestId in Body Parameters
Notification	Notify Coupon Scheme Changes (Including Add, Edit, Delete)	CouponScheme	CouponSchemeChangeNotify	v2.0: Route in Query Parameters v3.0: Route + RequestId in Body Parameters
Notification	Notify Control Lane Barrier Status (Including Release, Close, Normally)	--	BarrierStatusNotify	v2.0: Route is none v3.0: Route + RequestId

Interface Type	Interface Name	v2.0 Route Value	v3.0 Route Value	Field Difference (v3.0 vs v2.0)
	Release,Cancel Normally Release)			in Body Parameters
Notification	Notify Lane Popup Order Fee Modification	--	LaneFeeModifyNotify	v2.0: Route is none v3.0: Route + RequestId in Body Parameters
Notification	Notify Vehicle Payment Information	--	VehiclePaymentNotify	v2.0: Route is none v3.0: Route + RequestId in Body Parameters
Notification	Create Entry Order for Vehicle Without Plate	--	NoPlateEntryNotify	v2.0: Route is none v3.0: Route + RequestId in Body Parameters
Notification	Notify Callback Parameter Changes	--	CallbackParamChangeNotify	v2.0: Route is none v3.0: Route + RequestId in Body Parameters
Notification	Notify Failed Callback Business Retry	--	FailedCallbackRetryNotify	v2.0: Route is none v3.0: Route + RequestId in Body Parameters