

SMART PARKING 1.0
PARKING VACANCY API 1.0
SPECIFICATION
PART I: CORE APIS

VERSION: 1.0

21.07.2016

Version	Primary Author(s)	Description of Version	Date Completed
1.0	HKSARG	Final Version	21/7/2016

CONTENTS

1.	INTRODUCTION	2
1.1	PURPOSE	2
1.2	SCOPE	2
1.3	GLOSSARY	2
1.4	REFERENCES	2
2.	OVERVIEW	3
2.1	PUSH REQUEST	3
2.2	PULL SERVICE	4
3.	PARKING VACANCY API	5
3.1	AUTHENTICATION	5
3.1.1	<i>Construct the original URL of HTTP request</i>	5
3.1.2	<i>Append monotonically increasing timestamp</i>	6
3.1.3	<i>Append Access Key</i>	6
3.1.4	<i>Sign the message</i>	6
3.2	API	7
3.2.1	<i>Update parking vacancy data</i>	7
4.	VACANCY REST ENDPOINT	9
4.1	ENDPOINT REGISTRATION	9
4.1.1	<i>Endpoint Format</i>	9
4.1.2	<i>External System ID</i>	10
4.2	ENDPOINT DEFINITION	10

1. INTRODUCTION

1.1 Purpose

This document describes the Web Service APIs for uploading the parking vacancy data automatically to Smart Parking System and defines REST endpoint definition for external parties to invoke and implement.

1.2 Scope

This document is related to the release of Smart Parking System 1.0.

1.3 Glossary

Carpark Application – The external application that keep traces of vacant parking space. It can be cashier operated computer terminal or an automatic vehicle counting system located at the entrance and exits.

Carpark Server – The external server that is developed by carpark operators for collecting vacant parking space from different carparks in propriety way. It encapsulates parking vacancy data through REST endpoint defined by Smart Parking system.

Parking Server – The centralized server that collects parking vacancy data from different carpark servers and applications.

RESTful API – Web service APIs used for external applications to get and update parking vacancy data.

1.4 References

- HMAC: Keyed-Hashing for Message Authentication
<https://tools.ietf.org/html/rfc2104>
- The Use of HMAC-SHA-1-96 within ESP and AH
<https://tools.ietf.org/html/rfc2404>
- Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec
<https://tools.ietf.org/html/rfc4868>

2. OVERVIEW

Smart Parking System is designed to facilitate driver to find vacant parking space in view to reducing the traffic congestion and pollution. Parking Server is the core component of the system to store, manage and disseminate the carpark information (e.g. location, opening hours, charges, etc.) including real-time parking vacancy data uploaded from carparks.

Parking server supports two internet-based communication methods for carpark to upload parking vacancy data. **Push Request** is initialized by carpark application while **Pull Service** is initialized by parking server.

2.1 Push Request

Parking Vacancy API is a web service, which is primarily designed for proprietary carpark application sending vacancy data to parking server through the HTTP request directly. The details are described in Section 3. The carpark operators can randomly generate **Access Key** and **Access Secret** in our Administration Portal at <https://sps-op.pilotsmartke.gov.hk> for individual carpark. These pairs of access codes are used to sign the HTTP requests with HMAC-SHA256 for authentication purpose and ensure the message integrity. Figure 1 describes the flow on how the parking vacancy data is encapsulated in message and how to authenticate the incoming HTTP request.

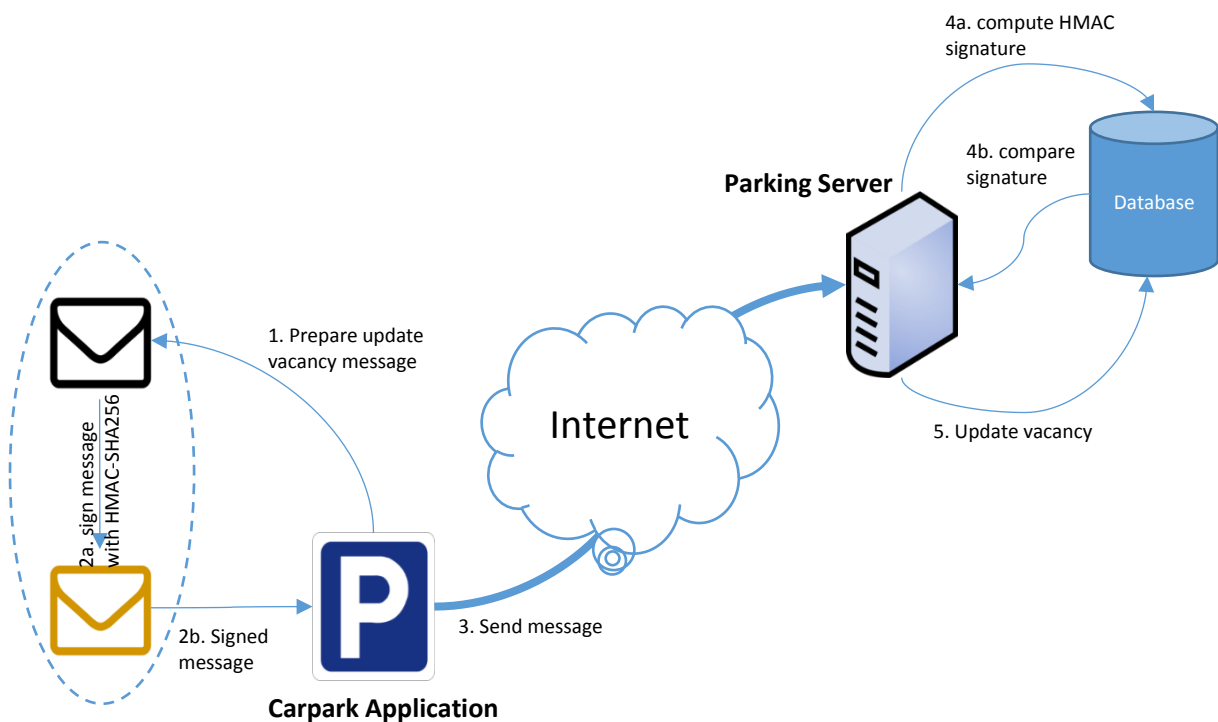


Figure 1. Push Request message flows

2.2 Pull Service

Pull service is designed for the carpark operators who have owned carpark server to centralize the parking vacancy data from different carpark terminals. This service allows them aggregating all parking vacancy data from different carparks in batch and being pulled by parking server periodically. The technical details are described in Section 4. Figure 2 describes the flow on how the parking vacancy data are aggregated by carpark server and how the parking server pulls the vacancy data to parking server periodically.

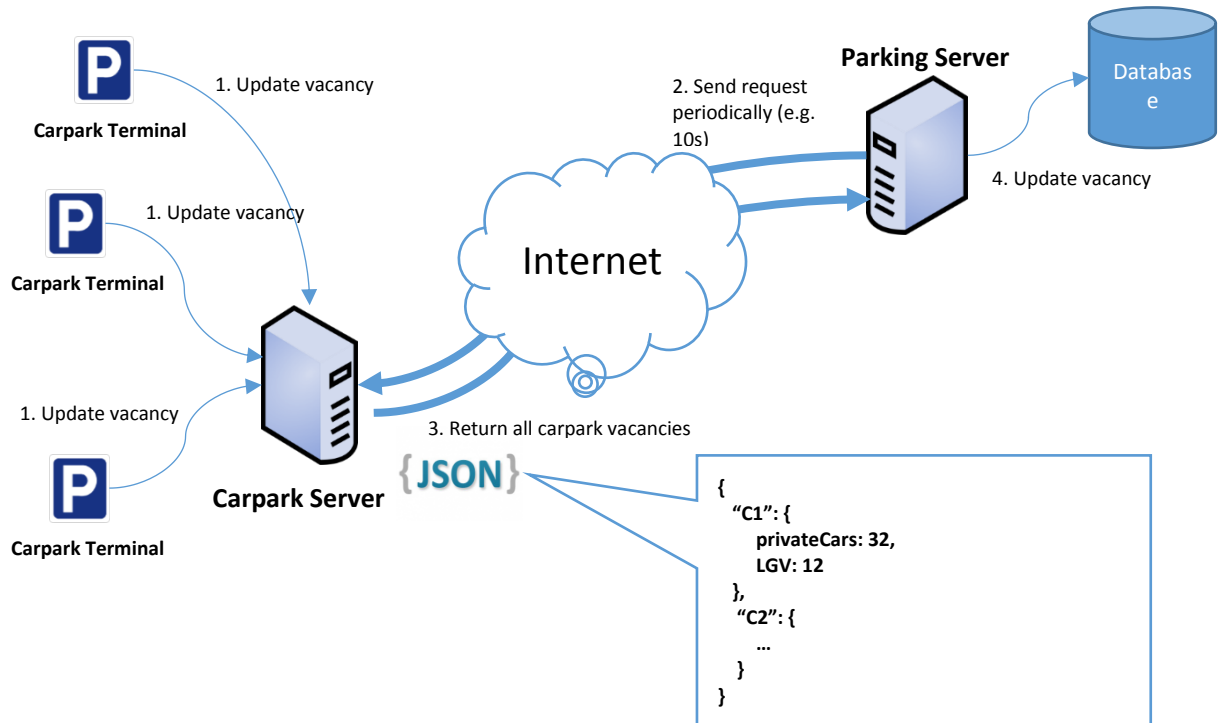


Figure 2. Pull Service message flows

3. PARKING VACANCY API

Parking Vacancy API is a RESTful web service API designed for remote invocation.

IMPORTANT NOTE! The hostname of Parking Vacancy API <https://sps-api.pilotsmartke.gov.hk> maybe changed as time evolving. It is recommended that the hostname is configurable when implementing the API.

3.1 Authentication

Parking server *would not* maintain sessions for HTTP requests from external party. External party must sign every HTTP request using HMAC-SHA256 signature with Access Secret. Once the HTTP request arrives to the parking server, the authentication manager will compute the signature using Access Secret looked-up from our database with the Access Key attached in the HTTP request. If the computed signature matches with the incoming signature from HTTP request, the parking server will accept the request for updating parking vacancy data. Otherwise, parking server would drop the request.

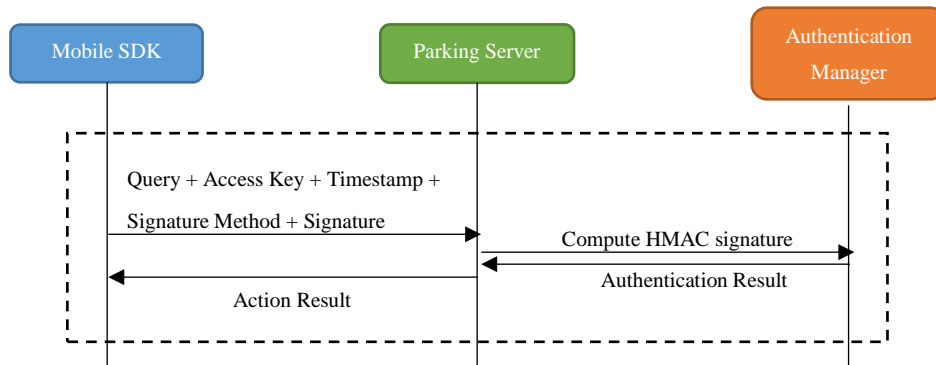


Figure 3. HMAC-SHA256 signature workflow

Though the HMAC-SHA256 signature method will prevent the Access Secret to be grabbed by malicious parties during transmission, they can perform a long run of an exhaustive brute-force attacks (i.e. 10^{48} times) to hit the Access Secret. Thus, it is recommended to change Access Key and Secret regularly in our Administration Portal at <https://sps-op.pilotsmartke.gov.hk>.

3.1.1 Construct the original URL of HTTP request

The original URL of HTTP request contains protocol, hostname, path and query parameters e.g. <https://sps-api.pilotsmartke.gov.hk/rest/updateVehicleVacancy?vehicleType=privateCar&vacancy=25>.

Our system only supports lowercase "https" as protocol scheme.

3.1.2 Append monotonically increasing timestamp

The HTTP request can be grabbed by malicious parties when sending over the Internet. They will attack our system by keep replaying the same HTTP request anytime. From view of this, the HTTP request shall append **timestamp** query parameter in the URL. For example:

```
https://sps-  
api.pilotsmartke.gov.hk/rest/updateVehicleVacancy?vehicleType=privateCar&vacancy=25&timestamp=1453284452796
```

The system clocks of carpark application are always assumed reasonably accurate and monotonically increasing. If any timestamp in HTTP request is **10 minutes** faster or slower than the system clock of our parking server, the system will drop the request directly. And the system will also drop any request if the timestamp is earlier than the last received timestamp of the same carpark in order to avoid out-of-order delivery of messages

3.1.3 Append Access Key

All HTTP requests must carry the Access Key, which identifies the carpark's credential. When the message arrives to the parking server, the authentication manager will compute the signature using Access Secret looked-up from our database with the Access Key attached in the HTTP request. The Access Key must be specified in **accessKey** query parameter.

```
https://sps-  
api.pilotsmartke.gov.hk/rest/updateVehicleVacancy?vehicleType=privateCar&vacancy=25&timestamp=1453284452796&accessKey=e91c2afd007a1950f68f7b7b2347a0aa2a0ef3a4
```

3.1.4 Sign the message

Once the URL string in the HTTP request is ready, it shall be signed with HMAC-SHA256. The signature method must be specified in **signatureMethod** query parameters. We currently only supported value is **sha256**, which represents HMAC-SHA256. We would explore to accept other signature method in future. The concatenated URL string becomes:

```
https://sps-  
api.pilotsmartke.gov.hk/rest/updateVehicleVacancy?vehicleType=privateCar&vacancy=25&timestamp=1453284452796&accessKey=e91c2afd007a1950f68f7b7b2347a0aa2a0ef3a4&signatureMethod=sha256
```

The URL string to be signed using HMAC is composed of original URL of the HTTP request, current timestamp, Access Key and signature method. By using this URL string to compute the signature with Access Secret and append the computed signature in **signature** query parameter. The final HTTP request becomes:

<https://sps-api.pilotsmartke.gov.hk/rest/updateVehicleVacancy?vehicleType=privateCar&vacancy=25×tamp=1453284452796&accessKey=e91c2afd007a1950f68f7b7b2347a0aa2a0ef3a4&signatureMethod=sha256&signature=406487c14877152abb34642dbd9bc5ea76f13efce731ea9d037cdac2fd344e20>

3.2 API

3.2.1 Update parking vacancy data

Update any parking vacancy data of various vehicle types including Private Car, Light/Heavy Good Vehicle, Coach and/or Motor Cycle.

Resource URL:

<https://sps-api.pilotsmartke.gov.hk/rest/updateVehicleVacancy>

Resource Information:

Response formats	JSON
Requires authentication	YES
Response compression	gzip/deflate supported
Response ETag	Supported

Parameters:

vehicleType	The type of vehicle, to which vacancies are updated. Possible values are privateCar , LGV , HGV , coach and motorCycle .
vacancy	Available parking space reserved for the specified type of vehicle. If parking spaces reserved for disabilities, electronic vehicle or unloading can be used inclusively for specified type of vehicle, this number <u>must</u> include these available parking spaces. Optional.
vacancyDIS	Available parking space reserved for disabled people driving specified type of vehicle. Optional.
vacancyEV	Available parking space reserved for specified type of electronic vehicle.

	Optional.
vacancyUNL	Available unloading space for specified type of vehicle. Optional.

Response:

Return all parking vacancy data of this vehicle type.

Example Request:

<https://sps-api.pilotsmartke.gov.hk/rest/updateVehicleVacancy?vehicleType=privateCar&vacancy=25&vacancyEV=3×tamp=1453284452796&accessKey=e91c2afd007a1950f68f7b7b2347a0aa2a0ef3a4&signatureMethod=sha256&signature=406487c14877152abb34642dbd9bc5ea76f13efce731ea9d037cdac2fd344e20>

Example Response:

```
{
  "vacancy": 25,
  "vacancyDIS": 3,
  "vacancyEV": 3,
  "vacancyUNL": 3
}
```

4. VACANCY REST ENDPOINT

The Vacancy REST endpoint defines the function that carpark server should implement. Parking server will call and retrieve latest parking vacancy data periodically from the carpark server.

4.1 Endpoint Registration

4.1.1 Endpoint Format

Carpark operator shall register the REST endpoint in Administration Portal at <https://sps-op.pilotsmartke.gov.hk>. Parking server will retrieve the parking vacancy data based on the REST endpoint which is a URL formulated by hostname and request URI. For example, you host the service on server www.example.com in relative path `/carpark/rest/getAllVacancies.jsp`. The complete endpoint is <http://www.example.com/carpark/rest/getAllVacancies.jsp>. Our system support both `http://` or `https://` protocols in the REST endpoint.

Figure 4. REST Endpoint registration

Our Parking server will use a fixed IP address to send HTTP POST requests. For security reason, carpark operators shall restrict the HTTP POST request from our fixed IP address only or assign an authentication token in the REST Endpoint URL. **(You may request us to provide the fixed IP address)**

4.1.2 External ID

Each carpark should have a unique identifier namely, **External ID** for identification purpose. The carpark operators should specify the External ID in Administration Portal The system will update the parking vacancy data according to the corresponding External ID.

The screenshot shows a web form titled 'General' for editing carpark information. The fields are as follows:

- ID:** 10
- *Name (in English):** Murray Road Car Park
- *Name (in Chinese):** 美利道停車場
- Nature:** Government
- Type:** Multi-storey Carpark Building
- Photo:** A photograph of a modern, multi-story carpark building with a glass facade and greenery in the foreground.
- External ID:** C01

Figure 5. Binding external ID to carpark

4.2 Endpoint Definition

The purpose of this REST endpoint is to get the all carpark vacancy data managed by the carpark operator.

Resource Information:

Response formats	JSON
Requires authentication	NO
Response compression	Optional
Response ETag	Optional

Parameters:

None.

Response:

The response is key value pairs of external carpark ID and all parking vacancy data reserved for any, disabilities, electronic vehicle or unloading of **FIVE** vehicle types including Private Car, Light/Heavy Good Vehicle, Coach and Motor Cycle.

Field Name	Type	Definition
{*}	String	The external ID of carpark.
privateCar	Number	Available parking space reserved for any private car. If parking spaces reserved for disabilities, electronic vehicle or unloading can be used inclusively for any private car, this number <u>must</u> include these available parking spaces. Optional.
privateCarDIS	Number	Available parking space reserved for disabled people driving a private car. Optional.
privateCarEV	Number	Available parking space reserved for electronic private cars. Optional.
privateCarUNL	Number	Available unloading space for private car. Optional.
LGV	Number	Available parking space reserved for any light goods vehicle. If parking spaces reserved for disabilities, electronic vehicle or unloading can be used inclusively for any light goods vehicle, this number <u>must</u> include these available parking spaces. Optional.
LGVDIS	Number	Available parking space reserved for disabled people driving a light goods vehicle. Optional.
LGVEV	Number	Available parking space reserved for electronic light goods vehicles. Optional.
LGVUNL	Number	Available unloading space for light goods vehicles. Optional.
HGV	Number	Available parking space reserved for any heavy goods vehicle. If parking spaces reserved for disabilities, electronic vehicle or unloading can be used inclusively for any heavy goods vehicle, this number <u>must</u> include these available parking spaces. Optional.
HGVDIS	Number	Available parking space reserved for disabled people driving a heavy goods vehicle. Optional.
HGVEV	Number	Available parking space reserved for electronic heavy goods vehicles. Optional.

HGVUNL	Number	Available unloading space for heavy goods vehicles. Optional.
coach	Number	Available parking space reserved for any coach. If parking spaces reserved for disabilities, electronic vehicle or unloading can be used inclusively for any motor cycle, this number <u>must</u> include these available parking spaces. Optional.
coachDIS	Number	Available parking space reserved for disabled people driving a coach. Optional.
coachEV	Number	Available parking space reserved for electronic coaches. Optional.
coachUNL	Number	Available unloading space for coaches. Optional.
motorCycle	Number	Available parking space reserved for any motor cycle. If parking spaces reserved for disabilities, electronic vehicle or unloading can be used inclusively for any motor cycle, this number <u>must</u> include these available parking spaces. Optional.
motorCycleDIS	Number	Available parking space reserved for disabled people driving a motor cycle. Optional.
motorCycleEV	Number	Available parking space reserved for electronic motor cycles. Optional.
motorCycleUNL	Number	Available unloading space for motor cycles. Optional.

Example Request:

<http://www.example.com/carpark/rest/getAllVacancies>

Example Response:

```
{
  "c01": {
    "privateCar":2,
    "privateCarDIS":2,
    "privateCarEV":2
  },
}
```

```
"C02": {  
  "privateCar":2,  
  "privateCarEV":2,  
  "LGV":0,  
  "LGVEV":0,  
  "motorCycle":0,  
  "motorCycleEV":0  
},  
"C03": {  
  "LGV":0,  
  "LGVEV":0  
}  
}
```