함께 만드는 서울, 함께 누리는 서울

서울특별시

서울시주차정보 OpenAPI 정의서

Revision 2.01

<2018-08-31>

서울특별시

문서 개정 내역

변 경 일	버전	변경 내용	작성자	승인자
2012-12-07	1.00	최초 작성		
2017-01-16	2.00	장애인차량/정기차량의 추가		

목차

1 서울시주차정보 OpenAPI란? 4

1.1 서울시주차정보 OpenAPI 구동방식 4

1.2 서울시주차정보 OpenAPI 서비스의 종류 4

1.2.1 주차장정보 OpenAPI 4

2 서울시주차정보 OpenAPI 명세 4

2.1 서울시주차정보 OpenAPI 기본 규약 4

2.2 주차장정보 OpenAPI 상세 5

2.2.1 주차장정보 OpenAPI 서비스 기능 6

2.2.1.1 주차장 정보 등록 요청 상세 6

2.2.1.2 주차장 운영 정보 등록(수정) 요청 상세 9

2.2.1.3 주차 현황 정보 등록(수정) 요청 11

2.2.1.4 주차면 상태 정보 등록(수정) 요청 12

2.2.1.5 주차 현황 정보 등록2(수정) 요청 12

1 서울시주차정보 OpenAPI란?

서울시주차정보 OpenAPI 플랫폼은 서울시통합주차정보시스템에서 제공하는 주차관련 정보를 웹, 데스크톱, 모바일 등 다양한 플랫폼에서 구동할 수 있도록 공개한 개발자 인터페이스 입니다. 본 문서는 서울시주차정보 OpenAPI 플랫폼을 사용하여 다양한 플랫폼과 서비스를 구성하는데 도움이 될 수 있도록 서울시주차정보 OpenAPI 플랫폼에서 제공하는 서비스의 종류, 정의, 규약 등을 명세하여 빠르고 손쉽게 서울시주차정보 OpenAPI 플랫폼을 사용할 수 있도록 지원하는데 있습니다.

1.1 서울시주차정보 OpenAPI 구동방식

서울시주차정보 OpenAPI는 웹 프로토콜(HTTP)로 호출되어 JSON(JavaScript Object Notation)을 사용하여 파라미터값을 수신하거나 결과 값을 전달합니다. 따라서 서울시주차정보 OpenAPI를 사용하는 애플리케이션은 웹 프로토콜을 이용하여 특정 URL로 요청을 전송하고, 기능에 따라 서버에서 요구되는 JSON 데이터를 생성하거나 서버로부터 전달되는 JSON 결과 데이터를 파싱해야 합니다.

1.2 서울시주차정보 OpenAPI 서비스의 종류

서울시통합주차정보 시스템의 서비스에 따라 서울시주차정보 OpenAPI는 크게 2가지로 구분되어 있습니다. "주차장 기본정보, 주차장 이용정보 서비스 구분이 이루어지며 각 서비스 별로 키 발급이 이루어져야 서비스가 가능합니다. 아래는 각 서비스 별 정의 내역입니다.

1.2.1 주차장정보 OpenAPI

주차장 정보, 운영현황, 주차현황 정보를 수집하는 서비스로 각 주차장 별로 운영중인 주차 관제 시스템과 연계되는 서비스로 구성되어 있습니다.

2 서울시주차정보 OpenAPI 명세

서울시주차정보 OpenAPI를 서비스 별로 구분하여 제공하는 기능의 종류와 내용을 상세하게 설명합니다.

2.1 서울시주차정보 OpenAPI 기본 규약

웹 프로토콜의 특성상 각 서비스 별 OpenAPI 는 데이터 송/수신용 REST 인터페이스 중 "POST"방식을 사용하여 데이터 송/수신이 이루어지며 데이터 송수신시 전달되는 파라미터의 기본 규약은 RFC 4627(JSON)를 사용합니다. 요청변수와 형식 등 기본 규약은 다음과 같습니다.

■ 한글 인코딩 : 한글은 utf-8 을 기준으로 URL Encoding 을 사용 합니다.

■ 요청 변수: 서버에 전송해야 하는 값으로, 요청 URL뒤로 변수명과 값을 'name=value' 형식으로 명시합니다.

■ 요청 변수 형식: 요청 변수에 따라 단순한 문자열 값이거나 JSON을 사용하여 서울시통합주차정보 시스템에서 사용하는 형식으로 작성된 값이어야 합니다.

■ 기본 에러 출력 메시지

Code	Message	Contents
0001	System error	시스템 에러
0010	Your query Request count is over the limit	쿼리요청 제한을 초과하였습니다. 일반적으로는 검색 API 25,000개, 지도 API 100,000개 이상의 쿼리에 대하여 이 에러 메시지가 발생되나, 쿼리 제한이 다르게 설정된 경우에는 이에 준하여 발생됩니다.
0011	Incorrect parameter Request	잘못된 파라미터 요청입니다. 각 메소드별로 설정되어야 하는 필드 자체가 없는 경우 발생하는 에러 메시지 입니다.
0020	Unregistered key	등록되지 않은 키입니다.
0021	Your key is temporary unavailable	사용할 수 없는 키입니다. OpenAPI에 등록되었으나, 일시적으로 사용 중지된 키를 통하여 쿼리를 검색하는 경우 발생하는 에러입니다.
0100	Invalid code value	부적절한 code 값입니다. 필드 정의에 없는 code 값을 사용한 경우, 혹은 code 값이 없는 경우 발생하는 메시지입니다.
0200	Not match symmetric key cryptosystem	대칭 키 암호화 방식에 의거하여 infra_id 의 암호화가 일치하지 않습니다.
0990	Undefined error occurred	정의되지 않은 오류가 발생하였습니다.

Sample(JSON) : {"result_state":"0020", "message"="Unregistered key" }

2.2 주차장정보 OpenAPI 상세

주차장 정보 OpenAPI는 주차장 정보, 운영현황, 주차현황 정보를 수집하는 서비스로 각 주차장별로 운영중인 주차관제 시스템과 연계되는 서비스로 구성되어 있습니다.

2.2.1 주차장정보 OpenAPI 서비스 기능

주차장정보 OpenAPI 서비스의 기능 특성상 OpenAPI 접근 시 서울시통합주차정보 시스템에서 요구되는 정보를 JSON 으로 작성하여 OpenAPI 의 파라미터로 전달되어야 합니다. 주차장정보 API 서비스 기능은 아래 표와 같습니다. 이 서비스 중 통합주차정보 시스템에 데이터를 전송하여 정보가 기록되는 서비스인 주차장 정보등록(수정)과 주차장 운영정보등록(수정)은 infra_id 값을 MAC 인증 방식의 Seed128 알고리즘을 사용하여 대칭 키 암호화를 적용하여 전송하여야 합니다.

주차장정보 관련 API	메소드	설명
1)주차장 정보 등록(수정)	ParkWritePost	주차장 인프라 구성요소(주차면, 자전거 주차 유무, 장애인 전용 주차면 등)인 기본 정보에 대해 등록, 수정 API
2)주차장 운영정보 등록(수정)	ParkOperPost	주차장 이용 형태, 이용 요금, 결제 매체 등 운영 정보 등록, 수정 API
3)주차장 이용정보 등록(수정)	**ParkStatPost**	**주차장 이용정보 전송 API**
4)주차면 이용정보 등록(수정)	**ParkSpacePost**	**주차면 이용정보 전송 API**

2.2.1.1 주차장 정보 등록 요청 상세

주차장 정보 등록 API 를 사용하기 위한 규약, 변수에 대한 정의 등 다양한 플랫폼을 지원하기 위한 공용 규약을 명세합니다.

통합주차정보시스템상에 정보 기록이 이루어지므로 이를 검증하는 역할을 수행할 수 있도록 일부 데이터를 암호화 하여 전송되어야 합니다.
암호화 방식은 SEED 128bit 방식을 사용하며, 대칭 키 방식으로 필수 필드인 parking_lot_name 필드 전위 Max 8Bytes, capacity 필드 전위 Max 4Byte, sync_time 필드 후위 Max 4Byte 를 조합하여 infra_id 값을 암호화 하여 전송합니다. 조합한 데이터의 크기가 16 이 안될 경우 Seed128 알고리즘 단위인 16 의 배수로 Padding 합니다. Padding 시 '0'(HEX:0x30,DEC:48)을 사용합니다.
암호화 상세 절차는 아래와 같습니다.

가) parking_lot_name 의 전위 8 자리 + capacity 전위 4 자리 + sync_time 후위 4 자리를 더하여 MAC 16Byte 값을 생성한다.

ex) parking_lot_name(주차장명): 세종로(6Byte)
capacity(주차면): 124(3Byte)
sync_time(동기화 시각): 2013-03-01 10:00:20.320(후위 4Byte)

MAC: { 0xBC, 0xBC, 0xC1, 0xBE, 0xB7, 0xCE, 0x31, 0x32, 0x34, 0x2E, 0x33, 0x32, 0x30, 0x30, 0x30, 0x30 }

주차장명이 8 자리가 안되므로 6 자리 + 주차면이 4 자리가 안되므로 3 자리 + 동기화 시각 후위 4 자리 = 13 Byte

Seed128 의 암복호화 기준 크기가 16 자리이므로 모자라는(16-13=3) 3 자리(붉은색으로 표시된 0x30, 0x30, 0x30)는 '0'(HEX:0x30, DEC:48)로 Padding 하여 생성 합니다.

나) 연계 신청시 수령한 비밀키로 생성한 MAC 16Byte(가.에서 생성한 값)를 암호화 합니다.

다) infra_ID 값을 MAC 16Byte 로 암호화 한 값(나.에서 암호화된 값)으로 다시 암호화 합니다.

라) 다.에서 암호화된 16Byte 를 base64 로 인코딩하여 32Byte 로 변환합니다.

마) Infra_ID 에 암호화된 라.에서 변환한 값을 설정합니다.

ex) infra_id=A1fyhSvh%2B60IBJQo6fjmcQ%3D%3D

한글데이터 및 특수 문자는 URL Encoding 을 이용하여 전송하여야 합니다.

■ 요청 URL: http://parking.seoul.go.kr/WCFB/ParklandWRT.svc/ParkWritePost

■ 요청변수(Request parameter)

Request param	Value	Contents
key	string(필수)	서울시통합주차정보안내 시스템에서 등록한 키 값 입니다. 키 등록 시 서비스 별로 신청하여 수신한 키 값으로 주차장정보 OpenAPI 서비스를 신청하여 수신한 키 값 입니다.
infra_id	string(필수)	서울시통합주차시스템에서 관리하는 주차장 별 key 값으로 발급받은 parking infra identify 값을 설정합니다.
parking_lot_name	string(필수)	주차장 명을 설정합니다. 한글 32 글자, 영문 64 글자를 설정할 수 있습니다.
in_charge	string	주차장 관리자명을 설정합니다. 최대 설정할 수 있는 글자는 한글 16 글자, 영문 32 글자 입니다.
phone	string	주차장 대표 전화번호를 설정합니다. 15 글자 이내의 숫자 값만 유효합니다.
capacity	int(필수)	주차장의 주차면 숫자를 설정합니다. 4 자리 이내의 숫자 값으로 설정합니다.
bike_racks	int	자전거 랙 수를 설정합니다. 사용하지 않을 경우 '0'으로 설정합니다. 4 자리 이내의 숫자 값으로 설정합니다.
compact	int	경차전용 주차면 수를 설정합니다. 사용하지 않으면 '0'이며 4 자리 이내의 숫자 값으로 설정합니다.
women	int	여성전용 주차면 수를 설정합니다. 사용하지 않으면 '0'이며 4 자리 이내의 숫자 값으로 설정합니다.
disabled	int	장애인전용 주차면, 사용하지 않을 경우 '0'으로 설정합니다. 4 자리 이내의 숫자 값으로 설정합니다.
electcar	int	전기차 주차면 수를 설정합니다. 사용하지 않으면 '0'이며 4 자리 이내의 숫자 값으로 설정합니다.
nanumcar	int	나눔카 주차면 수를 설정합니다. 사용하지 않으면 '0'이며

		4 자리 이내의 숫자 값으로 설정합니다.
motorbike	int	이륜차 주차면 수를 설정합니다. 사용하지 않으면 '0'이며 4 자리 이내의 숫자 값으로 설정합니다.
bus	int	버스 주차면 수를 설정합니다. 사용하지 않으면 '0'이며 4 자리 이내의 숫자 값으로 설정합니다.
sync_time	string(필수)	동기화 기준 데이터로 주차장 정보 등록 수정 시 기록된 동기화 기준 데이터(형식: "yyyy-mm-dd hh24:mi:ss.mmm")
position_longitude	float	주차장 경도값 126.97796919
position_latitude	float	주차장 위도값 37.566535

Sample

접근 URL: http://parking.seoul.go.kr/WCFB/ParklandWRT.svc/ParkWritePost

전송 방식: POST

Parameters:

key=G35N6FFJTB&infra_id=A1fyhSvh%2B60IBJQo6fjmcQ%3D%3D&parking_lot_name=%EB%B0%9C%EC%82%B0%EC%A3%BC%EC%B0%A8%EC%9E%A5&in_charge=&phone=02-1313-1231&capacity=140&bike_racks=1&compact=2&women=3&disabled=0&electcar=0&nanumcar=0&garden=0&bus=0&sync_time=2013-01-08+13%3A24%3A19.543&position_longitude=126.97796919&position_lititude=37.566535

■ 출력결과(response field) - JSON

Object Name	Group (Object Name)	Contents
result_state		성공 시 "0000", 실패 시 오류코드 값이 설정 됩니다.
message		성공적인 정보 등록 완료 시에는 아무 값이 설정되지 않거나 Field 자체가 존재하지 않습니다. 단 값이 설정되었을 경우 주차장 정보 등록 실패이며 원인, 이유 등을 메시지로 설정되어 회신 됩니다.

Sample page

Success

{"result_state":"0000" }

Fail

{"result_state":"0020", "message"="Unregistered key" }

2.2.1.2 주차장 운영 정보 등록(수정) 요청 상세

주차장 운영 정보 등록 API 를 사용하기 위한 규약, 변수에 대한 정의 등 다양한 플랫폼을 지원하기 위한 공용 규약을 명세합니다.

암호화 방식은 주차장 정보 등록 요청 상세와 같습니다. 다만 MAC 생성 필드는 key 필드 전위 Max 8Bytes, weekday_begin_time 필드 전위 Max 4Bytes, sync_time 필드 후위 Max 4Byte 를 사용합니다. 그 절차는 아래와 같습니다.

가) **key**(연계키)의 전위 8 자리 + **weekday_begin_time** 전위 4 자리 + **sync_time** 후위 4 자리를 더하여 MAC 16Byte 값을 생성한다.

ex) key(연계키): G35N6FFJTB (8Byte)
weekday_begin_time(평일운영시작시각): 0900(4Byte)
sync_time(동기화 시각): 2013-01-08 13:24:19.543 (후위 4Byte)
MAC: { 0x47, 0x33, 0x35, 0x4E, 0x36, 0x46, 0x46, 0x4A, 0x30, 0x39, 0x30, 0x30, 0x2E, 0x35, 0x34, 0x33 }

나) 연계 신청시 수령한 비밀키로 생성한 MAC 16Byte(가.에서 생성한 값)를 암호화 합니다.
다) infra_ID 값을 MAC 16Byte 로 암호화 한 값(나.에서 암호화된 값)으로 다시 암호화 합니다.
라) 다.에서 암호화된 16Byte 를 base64 로 인코딩하여 32Byte 로 변환합니다.
마) infra_ID 에 암호화된 라.에서 변환한 값을 설정합니다.

ex) infra_id=A1fyhSvh%2B60IBJQo6fjmcQ%3D%3D

한글데이터 및 특수 문자는 URL Encoding 을 이용하여 전송하여야 합니다.

- 요청 URL: http://parking.seoul.go.kr/WCFB/ParklandWRT.svc/ParkOperPost
- 요청변수(Request parameter)

Request param	Value	Contents
Key	string(필수)	서울시통합주차정보안내 시스템에서 등록한 키 값 입니다. 키 등록 시 서비스 별로 신청하여 수신한 키 값으로 주차장정보 OpenAPI 서비스를 신청하여 수신한 키 값 입니다.
Infra_id	string(필수)	서울시통합주차시스템에서 관리하는 주차장 별 key 값으로 발급받은 parking infra identify 값을 설정합니다.
Visitor_access	string(필수)	(필수) 외부 차량 주차 가능 여부(주차가능:Y, 입주자 전용:N)
position_class	string	서울특별시 기준 급지(1~5)
night_free_open	string	야간 개방 사업 여부(Y or N)
weekday_begin_time	string(필수)	(필수) 평일 운영 시작 시각(HHMM) 분 단위

weekday_end_time	string(필수)	(필수) 평일 운영 종료 시각(HHMM) 분 단위
weekend_begin_time	string	주말 운영 시작 시각(HHMM) 분 단위
weekend_end_time	string	주말 운영 종료 시각(HHMM) 분 단위
holiday_begin_time	string	공휴일 운영 시작 시각(HHMM) 분 단위
holiday_end_time	string	공휴일 운영 종료 시각(HHMM) 분 단위
rates	int (필수)	(필수) 기본 주차 요금
time_rate	int (필수)	(필수) 기본 요금 부과 시간(분 단위)
add_rates	int (필수)	(필수) (추가)주차 단위 요금
add_time_rate	int (필수)	(필수) (추가)주차 단위 요금 시간(분 단위)
day_maximum	int (필수)	(필수) 일 최대 요금
day_monthly	int	주간 정기권 요금
night_monthly	int	야간 정기권 요금
all_day_monthly	int	전일(주-야간) 정기권 요금
no_driving_rotation_system	int	부제 운영 코드(표 4. 참조)
compact_car	int	경차 할인율(백분율)
disabled	int	장애인 할인율(백분율)
national_merit	int	국가유공자 할인율(백분율)
many_children	int	다자녀 할인율(백분율)
store_discount	string	기타 할인 제공 여부 (ex. 주변상가할인,영화관,빌딩입주점포등) Y= 할인 제공, N= 할인 없음
notify_memo	string	추가 할인 안내 및 주차장 안내용 메모 (최대 영문 256 자, 한글 128 자)
sync_time	string(필수)	동기화 기준 데이터로 주차장 정보 등록 수정 시 기록된 동기화 기준 데이터 (형식: "yyyy-mm-dd hh24:mi:ss.mmm")

■ 부제 코드

부제코드	내용
1	부제 운영 안 함
2	2 부제
3	요일 제, 5 부제
4	10 부제
5	서울시승용차요일제

Sample

접근 URL: http://parking.seoul.go.kr/WCFB/ParklandWRT.svc/ParkOperPost

전송 방식: POST

Parameters:

key=G35N6FFJTB&infra_id=lKEIGTHse%2B9yLsEQ28suUA%3D%3D&visitor_access=N&position_class=1&night_free_open=Y&weekday_begin_time=0800&weekday_end_time=2230&weekend_begin_time=0900&weekend_end_time=2140&holiday_begin_time=0830&holiday_end_time=2130&rates=1000&time_rate=30&add_rates=500&add_time_rate=10&day_maximum=12000&day_monthly=36000&night_monthly=42000&all_day_monthly=68000&no_driving_rotation_system=2&compact_car=50&disabled=20&national_merit=30&many_children=25&store_discount=Y&notify_memo=%EB%8B%A4%EC%96%91%ED%95%9C+%ED%95%A0%EC%9D%B8%EC%98%B5%EC%85%98%EC%9D%84+%EC%A0%9C%EA%B3%B5%ED%95%98%EA%B3%A0+%EC%9E%88%EC%8A%B5%EB%8B%88%EB%8B%A4.+%EC%98%A4%EC%84%B8%EC%9A%94.&sync_time=2013-01-08+13%3A24%3A19.543

■ 출력결과(response field) - JSON

Object Name	Group (Object Name)	Contents
result_state		성공 시 "0000", 실패 시 오류코드 값이 설정 됩니다.
message		성공적인 정보 등록 완료 시에는 아무 값이 설정되지 않거나 Field 자체가 존재하지 않습니다. 단 값이 설정되었을 경우 주차장 운영 정보 등록 실패이며 원인, 이유 등을 메시지로 설정되어 회신 됩니다.

Sample page

Success

{"result_state":"0000" }

Fail

{"result_state":"0020", "message"="Unregistered key" }

2.2.1.3 주차장 이용정보 등록(수정) 요청

주차 현황 정보 등록 API 를 사용하기 위한 규약, 변수에 대한 정의 등 다양한 플랫폼을 지원하기 위한 공용 규약을 명세합니다.

■ 요청 URL: http://parking.seoul.go.kr/WCFB/ParklandWRT.svc/ParkStatPost

■ 요청변수(Request parameter)

Request param	Value	Contents
key	string(필수)	서울시통합주차정보안내 시스템에서 등록한 키 값 입니다. 키 등록 시 서비스 별로 신청하여 수신한 키

		값으로 주차장정보 OpenAPI 서비스를 신청하여 수신한 키 값 입니다.
infra_id	string(필수)	서울시통합주차시스템에서 관리하는 주차장 별 key 값으로 발급받은 parking infra identify 값을 설정합니다.
current_parking	string(필수)	(필수) 현재 주차 차량 수

Sample

접근 URL: http://parking.seoul.go.kr/WCFB/ParklandWRT.svc/ParkStatPost

전송 방식: POST

Parameters: key=G35N6FFJTB&infra_id=171702&current_parking=127

■ 출력결과(response field) - JSON

Object Group Name	(Object Name) Contents
result_state	성공 시 "0000", 실패 시 오류코드 값이 설정 됩니다.
message	성공적인 정보 등록 완료 시에는 아무 값이 설정되지 않거나 Field 자체가 존재하지 않습니다. 단 값이 설정되었을 경우 주차 현황 정보 등록 실패이며 원인, 이유 등을 메시지로 설정되어 회신 됩니다.

Sample page

Success

{"result_state":"0000" }

Fail

{"result_state":"0020", "message"="Unregistered key" }

2.2.1.4 주차면 상태 정보 등록(수정) 요청

주차면 상태 정보 등록(수정) API 를 사용하기 위한 규약, 변수에 대한 정의 등 다양한 플랫폼을 지원하기 위한 공용 규약을 명세합니다.

■ 요청 URL: http://parking.seoul.go.kr/WCFB/ParklandWRT.svc/ParkSpacePost

■ 요청 변수(Request parameter)

Request param	Value	Contents
key	string(필수)	서울시통합주차정보안내 시스템에서 등록한 키 값 입니다. 키 등록 시 서비스 별로 신청하여 수신한 키 값으로 주차장정보 OpenAPI 서비스를 신청하여 수신한 키 값 입니다.

infra_id		string(필수)	서울시통합주차시스템에서 관리하는 주차장 별 key 값으로 발급받은 parking infra identify 값을 설정합니다.
count		int(필수)	주차장의 주차면 숫자를 설정합니다. 4 자리 이내의 숫자 값으로 설정합니다.
List	floor_class	string	(반복)지상, 지하 구분(지상:F, 지하:B)
	floor_number	int	(반복)지상, 지하층 번호
	parking_space_id	string	(반복)주차면 번호
	parking_space_status	string	(반복)주차면 상태 (I:주차 중 주차면, O:빈 주차면)

※ Floor class & Floor number 설정 예제

	Floor class	Floor number	
지상 2층:	F	2	
지하 4층:	B	4	
지상 1층:	F	1	(노외 주차장 동일)

Sample

접근 URL: http://parking.seoul.go.kr/WCFB/ParklandWRT.svc/ParkSpacePost

전송 방식: POST

Parameters:

key=G35N6FFJTB&infra_id=171702&count=40&floor_class=F&floor_class=B&floor_class=&floor_number=1&floor_number=2&floor_number=&parking_space_id=10003&parking_space_id=10004&parking_space_id=10005

■ 출력 결과(response field) - JSON

Object Name	Group (Object Name)	Contents
result_state		성공 시 "0000", 실패 시 오류코드 값이 설정 됩니다.
message		성공적인 정보 등록 완료 시에는 아무 값이 설정되지 않거나 Field 자체가 존재하지 않습니다. 단 값이 설정되었을 경우 주차장 운영 정보 등록 실패이며 원인, 이유 등을 메시지로 설정되어 회신 됩니다.

Sample page

Success

{"result_state":"0000" }

Fail

{"result_state":"0020", "message"="Unregistered key" }

2.2.1.5 주차장 이용정보2 등록(수정) 요청(미사용)

주차 현황 정보 등록 API 를 사용하기 위한 규약, 변수에 대한 정의 등 다양한 플랫폼을 지원하기 위한 공용 규약을 명세합니다.

■ 요청 URL: http://parking.seoul.go.kr/WCFB/ParklandWRT.svc/ParkStatPost2

■ 요청변수(Request parameter)

Request param	Value	Contents
key	string(필수)	서울시통합주차정보안내 시스템에서 등록한 키 값 입니다. 키 등록 시 서비스 별로 신청하여 수신한 키 값으로 주차장정보 OpenAPI 서비스를 신청하여 수신한 키 값 입니다.
infra_id	string(필수)	서울시통합주차시스템에서 관리하는 주차장 별 key 값으로 발급받은 parking infra identify 값을 설정합니다.
current_parking	string(필수)	(필수) 현재 전체 주차 차량 수
disabled_parking	string(선택)	(선택) 현재 장애인전용 주차면 주차 차량 수
regular_parking	string(선택)	(선택) 현재 정기차량전용 주차면 주차 차량 수
nanum_parking	string(선택)	(선택) 현재 나눔카전용 주차면 주차 차량 수
elect_parking	string(선택)	(선택) 현재 전기차전용 주차면 주차 차량 수
garden_parking	string(선택)	(선택) 현재 이륜차전용 주차면 주차 차량 수
inout_gubun	String(선택)	(선택) 현재 발생한 변동 차량이 입차(I)/출차(O)
car_num	String(선택)	(선택) 현재 발생한 변동 차량번호 또는 발권번호
occur_time	String(선택)	(선택) 현재 발생한 변동 차량 입차(I)/출차(O) 시간 YYYYMMDDHHNNSS

Sample

접근 URL: http://parking.seoul.go.kr/WCFB/ParklandWRT.svc/ParkStatPost2

전송 방식: POST

Parameters:

key=G35N6FFJTB&infra_id=171702&current_parking=127&disabled_parking=12&regular_parking=27&nanum_parking=7& elect_parking=0& garden_parking& inout_gubun=O& car_num=서울00가1234&occur_time=20180101122334

■ 출력결과(response field) - JSON

Object	Group	(Object Name)	Contents

Name	
result_state	성공 시 "0000", 실패 시 오류코드 값이 설정 됩니다.
message	성공적인 정보 등록 완료 시에는 아무 값이 설정되지 않거나 Field 자체가 존재하지 않습니다. 단 값이 설정되었을 경우 주차 현황 정보 등록 실패이며 원인, 이유 등을 메시지로 설정되어 회신 됩니다.

Sample page

Success

{"result_state":"0000" }

Fail

{"result_state":"0020", "message"="Unregistered key" }