

**上海市停车平台停车场库
数据通信协议
(讨论稿)**

2019-10-31

数据通信协议

协议框架

市公共停车信息平台数据接入服务接口为 REST 服务，提供数据接入、权限认证、配置管理及相关查询功能。接口服务数据格式为 json 格式，数据编码为 UTF-8, 具体方式如下：

数据格式	HTTP head 参数	值	说明
Json	Accept	application/json	请求返回为 json 格式
	Content-Type	application/json	请求数据为 json 格式

接口调用完整 url 为：

[http://\\${serviceIp}:\\${port}/service/parking/\\${path}?\\${appId}=xxxx
&\\${nonce}=xxxx&\\${curTime}=xxxx&\\${checksum}=xxxx](http://${serviceIp}:${port}/service/parking/${path}?${appId}=xxxx&${nonce}=xxxx&${curTime}=xxxx&${checksum}=xxxx)

或

[https://\\${serviceIp}:\\${port}/service/parking/\\${path}?\\${appId}=xxx
x&\\${nonce}=xxxx&\\${curTime}=xxxx&\\${checksum}=xxxx](https://${serviceIp}:${port}/service/parking/${path}?${appId}=xxx
x&${nonce}=xxxx&${curTime}=xxxx&${checksum}=xxxx)

地址参数：

参数	参数说明
serviceIp	服务机器 IP 地址或机器名
port	服务端口号
path	数据服务类型
params	参数格式为 key=value 方式，多个参数以&符号

请求参数说明：

参数	参数说明
appId	平台分配的 appId

nonce	随机数（最大长度 128 个字符）
curTime	当前 UTC 时间戳, 从 1970 年 1 月 1 日 0 点 0 分 0 秒开始到现在的秒数(String)
checkSum	SHA1(password + nonce + curTime), 三个参数拼接的字符串, 进行 SHA1 哈希计算, 转化成 16 进制字符(String, 小写) password 注册用户时系统分配的密码

所有收费终端设备接入平台前需由平台进行数据、安全及完整性监测。检测通过后设备向平台注册后方可接入平台。

停车场库接入前需向平台提交基础信息申请接入，平台审核接入申请，同意接入后会分配给相关接入信息，如下信息：

appId: 接入账号

password: 接入密码

parkingId: 场库标识

serverIp: 接口服务地址 IP 或域名

serverPort: 接口服务端口号

clientIp: 客户端允许连接的 ip 地址，非固定 IP 可不设置

请求结果格式

```
{
    "code": 0,
    "message": "success",
    "data": 数据体
}
```

code 为 0 操作成功, 其它值则为失败, 具体内容参见错误码说明。

停车场库数据接口

车辆入场

车辆进入停车场时，停车场实时上传车辆入场记录，数据内容包括交易序号、车牌、车辆类型、车道类型、停车类型、当前剩余泊位、当前系统时间、数据签名等信息。

服务接口

接口名称	车辆入场				
接口描述	如果发送失败需要重新发送，数据实时性要求不得延迟超过 30 秒				
请求命令	POST				
接口 URL	PATH	/data/parkplot/arrive/{parkingId}			
	参数	参数名	类型	说明	必选
		parkingId		场库编号	√
输入	<pre>{ "seq": "xxxxxxx", "plateId": "皖 AP1833", "vehicleType": 1, "laneType": 1, "freeBerth": 100, "parkType": 1, "dataTime": 1507863248063, "sign": "xxxxxxx" }</pre>				
输出	<pre>{ "code": 0, "message": "success" }</pre>				

	}
异常情况	结果 code 为 0 操作成功，其它值参考“错误码说明”
其他说明	

字段说明

序号	字段	签名 字段	类型	可为 空	说明
1.	seq		string	否	交易序列号，需唯一，最大长度32位 建议编码格式： 场库+扩展字段+时间+流水 时间格式：yyyyMMddHHmmss
2.	plateId	是	string	否	车牌
3.	vehicleType	是	number	否	车型 1-大型 2-中型 3-小型 9-其它
4.	laneType		number	否	1-人工 2-自动（ETC、无感等） 9-其它
5.	freeBerth	是	number	否	空余车位数
6.	parkType		number		停车类型 1-临停 2-包月 9-其它
7.	dateTime	是	long	否	入场时间：UTC时间戳，从1970年1月1日0点0分0秒开始到现在的毫秒

					数,例如: 1420123421000
8.	sign		string	否	签名结果

车辆离场

车辆驶出停车场时,需实时上传车辆出场记录,出场记录数据包括交易序号、车牌、车辆类型、车道类型、停车类型、停车时长、当前剩余泊位、支付金额、支付方式、当前系统时间、数据签名等数据。

服务接口

接口名称	车辆离场				
接口描述	如果发送失败需要重新发送,数据实时性要求不得延迟超过 30 秒				
请求命令	POST				
接口 URL	PATH	/data/parkplot/leave/{parkingId}			
	参数	参数名	类型	说明	必选
		parkingId		场库编号	√
输入	<pre> { "seq": "xxxxxxx", "plateId": "皖 AP1833", "vehicleType": 1, "laneType": 1, "parkingTime": 3600, "parkType": 1, "freePerth": 200, "payMoney": 5000, "payType": "wechat", </pre>				

	<pre> "dataTime":1507863248063, "sign":"XXXXXXX" } </pre>
输出	<pre> { "code":0, "message":"success" } </pre>
异常情况	结果 code 为 0 操作成功，其它值参考“错误码说明”
其他说明	

字段说明

序号	字段	签名 字段	类型	可为 空	说明
1.	seq		string	否	交易序列号，需唯一，最大长度32位 建议编码格式： 场库+扩展字段+时间+流水 时间格式：yyyyMMddHHmmss
2.	plateId	是	string	否	车牌号码
3.	vehicleType	是	number	否	车型 1-大型 2-中型 3-小型 9-其它
4.	laneType	是	number	否	1-人工 2-自动（ETC、无感等） 9-其它
5.	parkingTime	是	number	否	停车时长（单位：秒）
6.	parkType		number		停车类型 1-临停

					2-包月 9-其它
7.	freeBerth	是	number	否	空余车位数
8.	payMoney	是	number	否	付款金额（单位：分）
9.	payType		string	否	wechat: 微信 alipay: 支付宝 uppay: 银联 cash: 现金 tcard: 公交卡 unknown: 其它
10.	dateTime	是	long	否	离场时间： UTC时间戳，从1970年1月1日0点0分0秒开始到现在的毫秒数,例如： 1420123421000
11.	sign		string	否	签名结果

心跳检测

停车场需定时上传心跳信息,心跳内容包括当日累计进场车辆数、当日累计离场车辆数、当前剩余泊位数、当前系统时间、数据签名等数据。平台收到心跳数据后返回当前平台时间,停车场库收到心跳结果,获取服务器时间,比较本地系统时钟,确保本地时间偏移少于1分钟。停车场库心跳发送周期为每五分钟一次。

服务接口

接口名称	心跳检测
接口描述	每5分钟发送一次
请求命令	POST

接口 URL	PATH	/manage/parkplot/heartbeat/{parkingId}			
	参数	参数名	类型	说明	必选
		parkingId		场库编号	√
输入	<pre> { "totalArrived":123, "totalLeft":321, "freeBerth":111, "dataTime":1420123421000, "sign":"XXXXXXXX" } </pre>				
输出	<pre> { "code":0, "message":"success", "data":{ "serverTime":1420123421000, "sign":"XXXXXXXX" } } </pre>				
异常情况	结果 code 为 0 操作成功，其它值参考“错误码说明”				
其他说明					

字段说明

序号	字段	签名 字段	类型	为空	说明
1.	totalArrived	是	number	否	日累计进场数
2.	totalLeft	是	number	否	日累计离场数
3.	freeBerth	是	number	否	当前空余车位数
4.	dataTime	是	number	否	数据时间UTC时间戳，从1970年1

					月1日0点0分0秒开始到现在的毫秒数,例如: 1420123421000
5.	<code>serverTime</code>	是	number	否	服务器返回时间: 当前UTC时间戳, 从1970年1月1日0点0分0秒开始到现在的毫秒数,例如: 1420123421000
6.	<code>sign</code>		string	否	签名结果

数据补传

停车场库需对所有漏传数据进行补传,补传时间为次日凌晨1点前,及次日凌晨1点前需确保前日数据完整性。

服务接口

接口名称	数据补传				
接口描述	当发现停车平台与停车场库记录不一致时,停车场库需对漏传所有进出场记录进行重新提交。				
请求命令	POST				
接口 URL	PATH	/data/resend/{parkingId}			
	参数	参数名	类型	说明	必选
		parkingId		场库编号	√
输入	<pre>{ "arrive": [{ "seq": "xxxxxxx", "plateId": "皖 AP1833", "vehicleType": 1, "laneType": 1, }] }</pre>				

	<pre> "freeBerth":100, "parkType":1, "dateTime":1507863248063, "sign":"XXXXXXX" }, ], "leave":[{ "seq":"xxxxxxx", "plateId":"皖 AP1833", "vehicleType":1, "laneType":1, "parkingTime":3600, "parkType":1, "freePerth":200, "payMoney":5000, "payType":"wechat", "dateTime":1507863248063, "sign":"XXXXXXX" }, ] } </pre>
输出	<pre> { "code":0, "message":"success" } </pre>
异常情况	结果 code 为 0 操作成功，其它值参考“错误码说明”

其他说明	
------	--

字段说明

序号	字段	签名 字段	类型	可为 空	说明
1.	arrive		List		入场记录列表
2.	seq		string	否	交易序列号，需唯一，最大长度32位 建议编码格式： 场库+扩展字段+时间+流水 时间格式：yyyyMMddHHmmss
3.	plateId	是	string	否	车牌
4.	vehicleType	是	number	否	车型 1-大型 2-中型 3-小型 9-其它
5.	laneType		number	否	1-人工 2-自动（ETC、无感等） 9-其它
6.	freeBerth	是	number	否	空余车位数
7.	parkType		number		停车类型 1-临停 2-包月 9-其它
8.	dateTime	是	long	否	入场时间：UTC时间戳，从1970年1月1日0点0分0秒开始到现在的毫秒数，例如：1420123421000
9.	sign		string	否	签名结果

序号	字段	签名 字段	类型	可为 空	说明
1.	leave		List		离场记录列表
2.	seq		string	否	交易序列号，需唯一，最大长度32位 建议编码格式： 场库+扩展字段+时间+流水 时间格式：yyyyMMddHHmmss
3.	plateId	是	string	否	车牌号码
4.	vehicleType	是	number	否	车型 1-大型 2-中型 3-小型 9-其它
5.	laneType	是	number	否	1-人工 2-自动（ETC、无感等） 9-其它
6.	parkingTime	是	number	否	停车时长（单位：秒）
7.	parkType		number		停车类型 1-临停 2-包月 9-其它
8.	freeBerth	是	number	否	空余车位数
9.	payMoney	是	number	否	付款金额（单位：分）
10.	payType		string	否	wechat: 微信 alipay: 支付宝 uppay: 银联 cash: 现金

					tcard: 公交卡 unknown: 其它
11.	dateTime	是	long	否	离场时间: UTC时间戳, 从1970年1月1日0点0分0秒开始到现在的毫秒数, 例如: 1420123421000
12.	sign		string	否	签名结果

预约信息下载

停车场库可通过后台接口查询场库预约信息, 每次获取预约全量数据, 场库获取数据后需处理 (修改、删除、添加) 的记录。建议每 10 分钟请求一次完成预约数据下载。

服务接口

接口名称	预约信息查询				
接口描述	预约信息查询				
请求命令	POST				
接口 URL	PATH	/query/reserved/{ parkingId}			
	参数	参数名	类型	说明	必选
		parkingId		场库编号	√
输入	无				
输出	<pre>{ "code": 0, "message": "success", "data": { "parkingId": "pd001", "reservations": [</pre>				

	<pre> { "plateId": "皖 AP1123", "reservedDay": "2019-11-06", "reservedType": 1, "vehicleType": 1, "driverName": "andyqin", "sign": "XXXXXXXX" }, { "plateId": "皖 AP1121", "reservedDay": "2019-11-07", "reservedType": 1, "vehicleType": 1, "driverName": "andyqin", "sign": "XXXXXXXX" }, { "plateId": "皖 AP1124", "reservedDay": "2019-11-08", "reservedType": 1, "vehicleType": 1, "driverName": "andyqin", "sign": "XXXXXXXX" }] } </pre>
异常情况	结果 code 为 0 操作成功，其它值参考“错误码说明”
其他说明	

字段说明

序号	字段	签名 字段	类型	为空	说明
1.	parkingId		string	否	停车场编号
2.	reservations		list	否	预约列表
3.	plateId	是	string	否	车牌
4.	reservedDay	是	Date	否	预约停车时间yyyy-MM-dd
5.	reservedType	是	number		1- 上午 2- 下午 3- 全天
6.	vehicleType	是	number		1-小车 2-大车
7.	driverName		string		驾驶员名称
8.	sign		string	否	签名结果

附录

错误码说明

序号	错误码	说明
1	1001	无效或不合法的 appId
2	1002	该 appId 无请求权限
3	1003	该 appId 已过期
4	1004	该 appId 被禁用
5	1005	该 appId 未启用
6	1006	不合法的参数或缺少必要参数
7	1007	请求参数校验错误
8	1008	被禁止的 IP 端访问

9	1009	无效的用户名和密码
10	1010	未登录的用户/终端
11	1011	未退出的用户/终端
12	1012	访问令牌已经过期
13	1013	无权限访问该服务
14	2001	请求过于频繁
15	2002	请求结果过大
16	2003	消息转发失败
17	2004	网络繁忙, 请稍后重试
18	2005	未知的请求类型
19	2006	不合法的请求
20	2007	内部服务器错误
21	2008	数据库表读取错误
22	2009	数据库表存储错误
23	2010	数据库表操作错误
24	2011	缓存数据库错误
25	2101	接口维护
26	2102	接口停用
27	2100	服务不可用
28	2900	内部服务器错误: {0}
29	2901	未知的服务器错误
30	3000	数据验证错误: {0}
31	3001	无效的枚举类型: {0}
32	3002	数据绑定失败: {0}
33	3003	数据验证错误: {0}
34	3004	不支持的数据内容类型: {0}
35	3005	不支持的请求方法类型: {0}
36	3006	无效的数据签名

常见问题解答

1. 常见错误码处理方式

◆ 1007 请求参数校验错误

答：请求 url 中参数 checksum 值错误，参考 Checksum 计算规则

<u>checksum</u>	SHA1 (password + nonce + <u>curTime</u>), 三个参数拼接的字符串, 进行 SHA1 哈希计算, 转化成 16 进制字符(String, 小写) password 注册用户时系统分配的密码
-----------------	---

◆ 3006 无效的数据签名

答：数据内容的签名字段(sign)值错误，参考 Sign 签名计算规则：

字段说明中标注有签名字段为“是”的字段，按照参数名 ASCII 码从小到大排序（字典序），将值连接起来，最后将分配的账号密码拼接到首部。之后将整个字符串（编码为 UTF-8）带入 MD5 签名算法中得出 sign 值

Checksum 计算

计算 CheckSum 的 java 代码举例如下：

```
import java.security.MessageDigest;

public class CheckSumBuilder {

    private static final char[] HEX_DIGITS = { '0', '1', '2', '3', '4', '5',
                                                '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f' };

    // 计算并获取CheckSum
```

```
public static String getChecksum(String appSecret, String nonce, String curTime)
{
    return encode("sha1", appSecret + nonce + curTime);
}
```

// 计算并获取md5值

```
public static String getMD5(String requestBody) {
    return encode("md5", requestBody);
}
```

```
private static String encode(String algorithm, String value) {
    if (value == null) {
        return null;
    }
    try {
        MessageDigest messageDigest = MessageDigest.getInstance(algorithm);
        messageDigest.update(value.getBytes());
        return getFormattedText(messageDigest.digest());
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
}
```

```
private static String getFormattedText(byte[] bytes) {
    int len = bytes.length;
    StringBuilder buf = new StringBuilder(len * 2);
    for (int j = 0; j < len; j++) {
        buf.append(HEX_DIGITS[(bytes[j] >> 4) & 0x0f]);
        buf.append(HEX_DIGITS[bytes[j] & 0x0f]);
    }
}
```

```
        return buf.toString();
    }
}
```

Sign 签名方式

字段说明中标注有签名字段为“是”的字段，按照参数名 ASCII 码从小到大排序（字典序），将值连接起来，最后将分配的账号密码（password）拼接到首部。之后带入 MD5 签名算法中得出 sign 验签

以离场接口示例，

Post 数据体：

```
{
    "seq": "pd00120190912001",
    "plateId": "皖 AP1855",
    "parkingTime": 3600,
    "vehicleType": 1,
    "freeBerth": 100,
    "parkType": 1,
    "laneType": 1,
    "payMoney": 50,
    "payType": "wechat",
    "dateTime": 1564648957258,
    "sign": "470afb503fbe6d3adefd5e5f6d3e0cfb"
}
```

按照签名字段定义，下列 plateId、vehicleType、parkingTime、freeBerth、payMoney、dateTime 为签名字段。按照参数名 ASCII 码

从小到大排序（字典序），将值连接起来：

`dataTime+freeBerth+parkingTime+payMoney+plateId+vehicleType`，

拼接数据内容为：1564648957258100360050 皖 AP18551，在起始位置加上预分配的 appId 对应的密码，最终需要签名数据如下(字符串编码 UTF-8)：

HWURVeVppkUOT20LvcoMhmjSaBkiKR1564648957258100360050 皖
AP18551

将上述数据进行 MD5 算法签名 (String, 小写)，获取签名字段内容：

sign= 816fe15325eff9a922550f409059717c