智慧停车停车场对接协议

（供内外部使用）

文档编号	ZW-0201
版 本 号	v3.0
作　　者	李鹏
日　　期	2020/11/05

杭州中威电子股份有限公司

修订历史记录

发布日期	版本	说明	作者
2020/11/05	V3.0	初稿	李鹏
2021/01/25	v3.1	增加停车场空余车位接口	李鹏

目录

1. 开发必读

1.1. 概述

本文档主要描述停车场数据对接接口服务 SDK(HTTP-API)，提供给三方上报停车场数据给运营系统和运营系统下发数据给停车场接口协议，具体包括认证方式，业务接口。

1.2. 术语定义

三方	上报停车场数据方
运营系统	接收停车场数据方
上行	三方向运营系统发起请求
下行	运营系统向三方发起请求

1.3. 协议规则

三方接入，调用 API 必须遵循以下规则：

传输方式	HTTP 传输
提交方式	采用 POST 方法提交
数据格式	请求与返回数据都为 JSON 格式
字符编码	统一采用 UTF-8 字符编码
签名算法	MD5 加密
签名要求	请求数据需要进行参数加签，详细方法请参考安全规范-签名算法
数据加密	采用 AES 加密方式
接入账号	运营系统提供
接入签名秘钥	运营系统提供
接入数据秘钥	运营系统提供

1.4. 返回码说明

详见返回码

1.5. 版本说明

接口标题后面的“【v1.0】”表示接口适用服务 v1.0 及以上版本。

2. 安全规范

2.1. 通用接口说明

☑ **概述**

所有对接进行的数据请求（上行/下行）使用的接口都是唯一的，主要业务接口区分是在 content 字段中。

☑ **请求**

请求网址　　系统部署好后运营系统给出

请求方法　　POST

请求类型　　application/json; charset=utf-8

请求参数	数据类型	长度	是否必填	描述
account	String	32	是	接入账号
timeStamp	String	32	是	当前时间戳，yyyyMMddHHmmss
sign	String	32	是	Md5(account,timestamp,signKey) 输 出小写字符串,其中 signKey 由运营系统提供;
content	String		是	请求信息主体 JSON 格式字符串，具体详见每个接口的示例说明

content

请求参数	数据类型	长度	是否必填	描述
method	String	32	是	接口定义里面的“请求名称”
param	String		是	接口定义的请求参数内容, JSON 字符串(对象转 JSON 字符串)

2.2. 加密规则

1、 数据加签：采用 md5 加密方式，将由账号、时间戳、加签秘钥组合进行加密处理;

```
/**
 * 生成签名串
 * 2018.9.25  kj  add
 * @param account      账号
 * @param timeStamp    时间戳，yyyyMMddHHmmss
 * @param signKey      私钥
 * @return    成功-签名串，失败-null
 */
public static String createSign(final String account, final String timeStamp, final String signKey) {
```

```
        //1.参数为空判断
        if (StringUtils.isBlank(account) || StringUtils.isBlank(signKey) || StringUtils.isBlank(timeStamp) ||
timeStamp.length() != 14) {
            return null;
        }

        //2.待 MD5 加密的字符串
        String tempStr = account + timeStamp + signKey;

        return VerifyUtil.encodeByMD5(tempStr).toLowerCase();//小写
}
```

2、数据验签

```
/**
 * 签名串校验
 * 2018.9.25  kj  add
 * @param account      账号
 * @param timeStamp    时间戳，yyyyMMddHHmmss
 * @param sign         待校验的签名串
 * @param signKey      私钥
 * @return     false-失败，true-成功
 */
public static boolean checkSign(final String account, final String timeStamp, final String sign, final String signKey) {
        //1.参数为空判断
        if (StringUtils.isBlank(account) || StringUtils.isBlank(sign) || StringUtils.isBlank(signKey)
                || StringUtils.isBlank(timeStamp) || timeStamp.length() != 14) {
            return false;
        }

        //2.待 MD5 加密的字符串
        String tempStr = account + timeStamp + signKey;

        //3.MD5 签名结果
        String tempSign = VerifyUtil.encodeByMD5(tempStr).toLowerCase();//小写

        return sign.equals(tempSign);
}
```

2.3. 对接账号

1、开发|测试对接信息
测试 signKey: 2c1743a391305fbf367df8e4f069f9f9
测试 dataKey: 91305fbf367df8e4
测试账号:alpha

测试地址: https://tingche.hzkting.com/openapi/v3/dclot

2、 正式对接信息

正式对接信息再正式上线后提供。

3. API 列表

3.1. 基础信息

3.1.1 停车场基础信息【v3.0】(上行)

停车场基础信息(parklotInfo)				
概述:	三方上报停车场基础信息，基础信息发生变化时上报一次，也可以定时 N 分钟上报一次，实际已项目需求来定，没有明确要求的时候建议定时 5 分钟一次；			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
parklotName	String	128	是	车场名称
address	String	256	是	车场地址
lat	double	12	否	车场纬度，高德地图
lng	double	12	否	车场经度，高德地图
contactName	String	32	否	联系人姓名
contactPhone	String	32	否	联系人电话
description	String	256	否	车场描述
regionList	Array		是	车场场区信息，空余泊位信息
+parklotCode	String	32	是	运营系统提供的车场编号
+regionId	int	11	是	场区 ID，只有一个的时候可以填写 1
+regionName	String	64	是	场区名称，建议和场区名称一致
+totalLots	int	11	是	场区总车位数
+currentLots	int	11	是	场区剩余车位数

请求示例代码:

```
private static void sendParklotInfo() throws Exception{
    String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
    String dataKey = "91305fbf367df8e4";
    String signKey = "2c1743a391305fbf367df8e4f069f9f9";
    String account = "alpha";

    List<Object> regionList = new ArrayList<>();
    Map<String, Object> regionMap = new HashMap<>();
    regionMap.put("parklotCode", "100001");
    regionMap.put("regionId", 1);
    regionMap.put("regionName", "测试停车场");
    regionMap.put("totalLots", 100);
    regionMap.put("currentLots", 55);
```

```
        regionList.add(regionMap);

        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("parklotCode", "100001");
        paramMap.put("parklotName", "测试停车场");
        paramMap.put("address", "余杭区衢海大厦地下车库");
        paramMap.put("lat", 30.0257);
        paramMap.put("lng", 119.936782);
        paramMap.put("contactName", "匿名");
        paramMap.put("contactPhone", "18888889999");
        paramMap.put("description", null);
        paramMap.put("regionList", regionList);

        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "parklotInfo");
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("http://test/openapi/v3/dclot", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","httpStatus":200,"data":1}

3.1.2 停车场车道基础信息【v3.0】(上行)

车道基础信息(parklotLane)	
概述：	以数组模式上传车道基础信息，用于后面做业务处理；车道信息发生变更时上报一次，或者每日定时上报一次；

接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
laneCode	String	32	是	车道编号
laneName	String	64	是	车道名称
laneType	int	2	是	车道类型编号

请求示例代码：

```
private static void sendParklotLaneList() throws Exception{
        String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
        String dataKey = "91305fbf367df8e4";
        String signKey = "2c1743a391305fbf367df8e4f069f9f9";
        String account = "alpha";

        List<Object> laneList = new ArrayList<>();
        Map<String, Object> inLaneMap = new HashMap<>();
        inLaneMap.put("parklotCode", "100001");
        inLaneMap.put("laneCode", "415F4E69D1648E580065_1");
        inLaneMap.put("laneName", "入口");
        inLaneMap.put("laneType", 1);
        Map<String, Object> outLaneMap = new HashMap<>();
        outLaneMap.put("parklotCode", "100001");
        outLaneMap.put("laneCode", "415F4E69D1648E580065_2");
        outLaneMap.put("laneName", "出口");
        outLaneMap.put("laneType", 2);
        laneList.add(inLaneMap);
        laneList.add(outLaneMap);

        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "parklotLane");
        contentMap.put("param", GsonUtil.toJson(laneList));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("http://test/openapi/v3/dclot", reqMap);

        System.out.println(json);
}
```

请求响应结果				
字段名	字段类型	长度	是否必填	描述

code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","httpStatus":200,"data":1}

3.2. 过车数据

3.2.1 车辆待入场【v3.0】(上行)

车辆待入场(passWaitIn)				
概述：	车辆入口压线未开闸时，上报待入场数据；			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
plate	String	32	是	车牌号码，若没有车牌则固定传“无车牌”
plateColor	String	2	是	车牌颜色编号
passTime	String	32	是	过车时间，yyyyMMddHHmmss
laneCode	String	32	是	车道编号
vehicleType	int	2	是	车辆类型编号
picVehicleFileData	String		否	过车图片，base64 字符串
picPlateFileData	String		否	车牌图片，base64 字符串

请求示例代码：

```
private static void sendWaitIn() throws Exception{
        String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
        String dataKey = "91305fbf367df8e4";
        String signKey = "2c1743a391305fbf367df8e4f069f9f9";
        String account = "alpha";

        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("parklotCode", "100001");
        paramMap.put("plate", "无车牌");
        paramMap.put("plateColor", "0");
        paramMap.put("passTime", "20201106123502");
        paramMap.put("laneCode", "415F4E69D1648E580065_1");
        paramMap.put("vehicleType", 1);
        paramMap.put("picVehicleFileData", null);
        paramMap.put("picPlateFileData", null);

        Map<String, String> contentMap = new HashMap<String, String>();
```

```
        contentMap.put("method", "passWaitIn");
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("http://test/openapi/v3/dclot", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			

注：
可入场：code=000000
不可入场：其他返回码

响应结果示例（无牌车）：
{"code":"020101","msg":"无牌车，禁止入场","httpStatus":200,"data":0}
响应结果示例（有牌车）：
{"code":"020103","msg":"待入场","httpStatus":200,"data":0}

3.2.2 车辆入场【v3.0】(上行)

车辆入场(passIn)				
概述:	入口车场开闸放行后，实时上报车辆入场数据			

接口请求参数

字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
uniqueNo	String	32	是	三方上报停车场入场唯一编号
plate	String	32	是	车牌号码
plateColor	String	2	是	车牌颜色编号
passTime	String	32	是	过车时间，yyyyMMddHHmmss

laneCode	String	32	是	车道编号
passTypeCode	String	2	是	放行方式编号
base64Pic	String		否	过车图片，base64 字符串
base64PlatePic	String		否	车牌图片，base64 字符串
opUser	String	32	是	收费员账号/姓名
vehType	int	2	是	车辆类型编号

请求示例代码：

```
private static void sendPassIn() throws Exception{
        String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
        String dataKey = "91305fbf367df8e4";
        String signKey = "2c1743a391305fbf367df8e4f069f9f9";
        String account = "alpha";

        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("parklotCode", "100001");
        paramMap.put("uniqueNo", "415F4E69D167590CD0B8C48");
        paramMap.put("plate", "浙 A88888");
        paramMap.put("plateColor", "0");
        paramMap.put("passTime", "20201106123502");
        paramMap.put("laneCode", "415F4E69D1648E580065_1");
        paramMap.put("passTypeCode", "50");
        paramMap.put("base64Pic", null);
        paramMap.put("base64PlatePic", null);
        paramMap.put("opUser", "admin");
        paramMap.put("vehType", 1);

        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "passIn");
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("http://test/openapi/v3/dclot", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000

msg	String	128	是	结果说明
data	Object			NULL
响应结果示例： **{"code":"000000","msg":"请求服务正常返回","httpStatus":200,"data":1}**				

3.2.3 车辆待离场【v3.0】(上行)

车辆待离场(passWaitOut)				
概述：	车辆离场压线未开闸时，上报待离场数据；			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
laneCode	String	32	是	车道编号
plate	String	32	是	车牌名称
plateColor	String	2	是	车牌颜色编号
passTime	String	32	是	过车时间，yyyyMMddHHmmss
vehicleType	int	2	是	车辆类型编号
picVehicleFileData	String		否	过车图片，base64 字符串
picPlateFileData	String		否	车牌图片，base64 字符串
shouldPay	int	11	是	离场时应付金额，分
请求示例代码：				

```
private static void sendWaitOut() throws Exception{
    String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
    String dataKey = "91305fbf367df8e4";
    String signKey = "2c1743a391305fbf367df8e4f069f9f9";
    String account = "alpha";

    Map<String, Object> paramMap = new HashMap<>();
    paramMap.put("parklotCode", "100001");
    paramMap.put("plate", "浙 A88888");
    paramMap.put("plateColor", "0");
    paramMap.put("passTime", "20201106164123");
    paramMap.put("laneCode", "415F4E69D1648E580065_2");
    paramMap.put("vehicleType", 1);
    paramMap.put("picVehicleFileData", null);
    paramMap.put("picPlateFileData", null);
    paramMap.put("shouldPay", 400);

    Map<String, String> contentMap = new HashMap<String, String>();
    contentMap.put("method", "passWaitOut");
    contentMap.put("param", GsonUtil.toJson(paramMap));
```

```
        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("http://test/openapi/v3/dclot", reqMap);

        System.out.println(json);
}
```

请求响应结果				
字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			0 或者 1

注：

可离场：code==000000 && data==0

不可离场：code=其他返回码 或者 code==000000 && data==1

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","httpStatus":200,"data":1}

3.2.4 车辆离场【v3.0】(上行)

车辆离场(passOut)				
概述：	开闸放行后，实时上报车辆离场数据；			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
uniqueNo	String	32	是	离场时停车场唯一记录编号
plate	String	64	是	车牌号码
plateColor	String	2	是	车牌颜色编号
passTime	String	32	是	过车时间，yyyyMMddHHmmss
laneCode	String	32	是	车道编号
passTypeCode	String	2	是	放行方式编号
extPassDesc	String	256	否	异常放行说明
base64Pic	String		否	过车图片，base64 字符串

base64PlatePic	String		否	车牌图片，base64 字符串
shouldPay	int	11	是	应付总停车费，分
actualPay	int	11	是	实付总停车费，分
inUniqueNo	String	32	否	离场记录对应的入场记录编号
opUser	String	32	是	收费员账号/姓名

请求示例代码：

```
private static void sendPassOut() throws Exception{
        String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
        String dataKey = "91305fbf367df8e4";
        String signKey = "2c1743a391305fbf367df8e4f069f9f9";
        String account = "alpha";

        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("parklotCode", "100001");
        paramMap.put("uniqueNo", "415F4E69D167590CD0B8B99");
        paramMap.put("plate", "浙 A88888");
        paramMap.put("plateColor", "0");
        paramMap.put("passTime", "20201106164123");
        paramMap.put("laneCode", "415F4E69D1648E580065_2");
        paramMap.put("passTypeCode", "50");
        paramMap.put("extPassDesc", null);
        paramMap.put("base64Pic", null);
        paramMap.put("base64PlatePic", null);
        paramMap.put("opUser", "admin");
        paramMap.put("shouldPay", 400);
        paramMap.put("actualPay", 400);
        paramMap.put("inUniqueNo", "415F4E69D167590CD0B8C48");

        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "passOut");
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("http://test/openapi/v3/dclot", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL
响应结果示例： **{"code":"000000","msg":"请求服务正常返回","httpStatus":200,"data":1}**				

3.2.5 重写无牌车车牌【v3.0】(下行)

重写无牌车车牌(rewriteNoPlateForWaitPass)				
概述：	将待入场和待离场时抓拍的无牌车改为虚拟车牌，虚拟车牌运营系统提供；			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
plate	String	64	是	运营系统虚拟的车牌号码
passTime	String	32	是	待入场/待离场车辆的过车时间，yyyyMMddHHmmss
laneCode	String	32	是	待入场/待离场记录所在车道编号
direction	int	2	是	方向，0-入场，1-离场
laneHostUrl	String	256	否	车道对应主机的地址信息（预留）

请求示例代码：

```
private static void sendRewriteNoPlate() throws Exception{
    String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
    String dataKey = "91305fbf367df8e4";
    String signKey = "2c1743a391305fbf367df8e4f069f9f9";
    String account = "alpha";

    Map<String, Object> paramMap = new HashMap<>();
    paramMap.put("parklotCode", "100001");
    paramMap.put("plate", "临 A10001");
    paramMap.put("passTime", "20201107101254");
    paramMap.put("laneCode", "415F4E69D1648E580065_1");
    paramMap.put("direction", 0);
    paramMap.put("laneHostUrl", null);

    Map<String, String> contentMap = new HashMap<String, String>();
    contentMap.put("method", "rewriteNoPlateForWaitPass");
    contentMap.put("param", GsonUtil.toJson(paramMap));

    String content = GsonUtil.toJson(contentMap);
```

```
        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("三方 URL", reqMap);

        System.out.println(json);
}
```

请求响应结果				
字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":1 }

3.2.6 删除场内车【v3.0】(下行)

删除场内车(parkingDelPlate)	
概述：	删除停车场在场的场内车记录；运营系统清理场内车业务场景使用；

接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
inUniqueNo	String	32	是	入场记录编号

请求示例代码：

```
private static void sendDelParkingInPlate() throws Exception{
        String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
        String dataKey = "91305fbf367df8e4";
        String signKey = "2c1743a391305fbf367df8e4f069f9f9";
        String account = "alpha";

        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("parklotCode", "100001");
        paramMap.put("inUniqueNo", "415F4E69D167590CD0B8C5E");

        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "parkingDelPlate");
        contentMap.put("param", GsonUtil.toJson(paramMap));
```

```
        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("三方 URL", reqMap);

        System.out.println(json);
}
```

请求响应结果				
字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL

响应示例代码：

{"code":"000000","msg":"请求服务正常返回","data":1}

3.3. 收费&自助缴费

3.3.1 过车收费记录【v3.0】(上行)

过车收费记录(parkingAlreadyPay)	
概述：	停车场自行收费后，实时上报过车收费记录； 注：运营系统支付的不需要再次上报，只需要上报停车场收费的所有记录；

接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
orderCode	String	32	是	停车场收费唯一记录编号
plate	String	32	是	车牌号码
plateColor	String	2	是	车牌颜色编号
payTime	String	32	是	支付时间，yyyyMMddHHmmss
shouldPay	int	11	是	应付金额，分
actualPay	int	11	是	实付金额，分

payType	String	2	是	支付方式编号
thirdSn	String	32	否	微信支付单号/支付宝支付单号，停车场扫码枪支付场景
inUniqueNo	String	32	否	离场记录对应的入场记录编号
opUser	String	32	是	收费员账号/姓名
isExtPass	int	2	是	离场是否是异常放行，0-否，1-是

请求示例代码：

```
private static void sendParkingAlreadyPay() throws Exception{
        String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
        String dataKey = "91305fbf367df8e4";
        String signKey = "2c1743a391305fbf367df8e4f069f9f9";
        String account = "alpha";

        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("parklotCode", "100001");
        paramMap.put("orderCode", "369843a391305fbf367df8e4");
        paramMap.put("plate", "浙 A88888");
        paramMap.put("plateColor", "0");
        paramMap.put("payTime", "20201106164123");
        paramMap.put("shouldPay", 400);
        paramMap.put("actualPay", 400);
        paramMap.put("payType", "0");
        paramMap.put("thirdSn", null);
        paramMap.put("opUser", "admin");
        paramMap.put("isExtPass", 0);
        paramMap.put("inUniqueNo", "415F4E69D167590CD0B8C48");

        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "parkingAlreadyPay");
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("http://test/openapi/v3/dclot", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述

code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL
响应结果示例： **{"code":"000000","msg":"请求服务正常返回","httpStatus":200,"data":1}**				

3.3.2 查询停车费-入场编号【v3.0】(下行)

查询停车费-入场编号(queryParkingFeeInUniqueNo)				
概述：	根据三方上报的入场记录中记录编号进行查询停车费；此接口用于无牌车出口扫码缴费业务处理；			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
inUniqueNo	String	32	是	停车场入场唯一记录编号
vehType	int	2	否	计费车辆类型编号，若不填，默认按小型车收费规则计费

请求示例代码：

```
private static void sendQueryParkingFeeInUniqueNo() throws Exception{
    String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
    String dataKey = "91305fbf367df8e4";
    String signKey = "2c1743a391305fbf367df8e4f069f9f9";
    String account = "alpha";

    Map<String, Object> paramMap = new HashMap<>();
    paramMap.put("parklotCode", "100001");
    paramMap.put("inUniqueNo", "415F4E69D167590CD0B8B85");

    Map<String, String> contentMap = new HashMap<String, String>();
    contentMap.put("method", "queryParkingFeeInUniqueNo");
    contentMap.put("param", GsonUtil.toJson(paramMap));

    String content = GsonUtil.toJson(contentMap);

    Map<String, String> reqMap = new HashMap<String, String>();
    reqMap.put("account", account);
    reqMap.put("timeStamp", timeStamp);
    reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
    reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

    String json = HttpPost.sendJsonObject("三方 URL", reqMap);
```

```
        System.out.println(json);
}
```

请求响应结果				
字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			查询结果对象
+plate	String	32	是	车牌号码
+preBillUuid	String	32	是	停车场预付账单号，用于后面支付成功通知接口使用
+parkingTime	int	11	是	停车时长，分钟
+shouldPay	int	11	是	当前应付金额，分
+billingStartTime	String	32	是	计费开始时间，yyyyMMddHHmmss
+billingEndTime	String	32	是	计费结束时间，yyyyMMddHHmmss
+inUniqueNo	String	32	是	停车场入场唯一记录编号

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":{"plate":"湘AKJ888","preBillUuid":"5ad9dc7867e941e8b5b32c71e178c4fa","parkingTime":"16851","shouldPay":0,"billingStartTime":"20201026202344","billingEndTime":"20201107131440","inUniqueNo":"415F4E69D167590CD0B8B85"},"httpStatus":200}

3.3.3 查询停车费-车牌号码【v3.0】(下行)

查询停车费-车牌号码(queryParkingFeePlate)				
概述：	根据车牌号码进行查询停车费；通用查询接口；			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
plate	String	32	是	车牌号码，正常识别出来的车牌号码
plateColor	String	2	是	车牌颜色编号
vehType	int	2	否	计费车辆类型编号，若不填，默认按小型车收费规则计费

请求示例代码：

```
private static void sendQueryParkingFeePlate() throws Exception{
        String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
        String dataKey = "91305fbf367df8e4";
        String signKey = "2c1743a391305fbf367df8e4f069f9f9";
        String account = "alpha";

        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("parklotCode", "100001");
        paramMap.put("plate", "湘 AKJ888");
        paramMap.put("plateColor", "0");
        paramMap.put("vehType", null);

        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "queryParkingFeePlate");
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("三方 URL", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			查询结果对象
+plate	String	32	是	车牌号码
+preBillUuid	String	32	是	停车场预付账单号，用于后面支付成功通知接口使用
+parkingTime	int	11	是	停车时长，分钟
+shouldPay	int	11	是	当前应付金额，分
+billingStartTime	String	32	是	计费开始时间，yyyyMMddHHmmss
+billingEndTime	String	32	是	计费结束时间，yyyyMMddHHmmss
+inUniqueNo	String	32	是	停车场入场唯一记录编号

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":{"plate":"湘AKJ888","preBillUuid":"01e5c17be54a4a89ae4d11dc0370484b","parkingTime":"16859","shouldPay":28690,"billingStartTime":"20201026202344","billingEndTime":"20201107132214","inUniqueNo":"415F4E69D167590CD0B8B85"},"httpStatus":200}

3.3.4 支付成功通知停车场【v3.0】(下行)

支付成功通知停车场(notifyParkingFee)				
概述：	运营系统缴费成功后，将支付结果下行通知停车场；			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
plate	String	32	是	车牌号码
preBillUuid	String	32	是	查询停车费时停车场给出的预付账单号
inUniqueNo	String	32	是	停车场入场记录编号
payFee	int	11	是	实际支付金额，分
payTime	String	32	是	支付时间，yyyyMMddHHmmss
payModel	int	2	是	支付模式
desc	String	128	否	缴费说明

请求示例代码：

```
private static void sendPaySuccNotify() throws Exception{
    String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
    String dataKey = "91305fbf367df8e4";
    String signKey = "2c1743a391305fbf367df8e4f069f9f9";
    String account = "alpha";

    Map<String, Object> paramMap = new HashMap<>();
    paramMap.put("parklotCode", "100001");
    paramMap.put("plate", "湘 AKJ888");
    paramMap.put("preBillUuid", "01e5c17be54a4a89ae4d11dc0370484b");
    paramMap.put("inUniqueNo", "415F4E69D167590CD0B8B85");
    paramMap.put("payFee", 28690);
    paramMap.put("payTime", DateUtil.format(new Date(), "yyyyMMddHHmmss"));
    paramMap.put("payModel", 1);
    paramMap.put("desc", "微信公众号支付");

    Map<String, String> contentMap = new HashMap<String, String>();
    contentMap.put("method", "notifyParkingFee");
    contentMap.put("param", GsonUtil.toJson(paramMap));
```

```
        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("三方 URL", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":1}

3.4. 会员车

3.4.1 添加/修改会员车【v3.0】(下行)

添加/修改会员车(addVehicle)				
概述：	若停车场已经存在此车则直接更新，若不存在做新增处理；			

接口请求参数

字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
plate	String	32	是	车牌号码
plateColor	String	2	是	车牌颜色编号
createTime	String	32	是	创建时间，yyyyMMddHHmmss
startTime	String	32	是	开始时间，yyyyMMddHHmmss
endTime	String	32	是	结束时间，yyyyMMddHHmmss
userName	String	32	否	车主姓名
userPhone	String	32	否	车主联系方式

请求示例代码：

```
private static void sendAddVehicle() throws Exception{
        String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
        String dataKey = "91305fbf367df8e4";
        String signKey = "2c1743a391305fbf367df8e4f069f9f9";
        String account = "alpha";

        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("parklotCode", "100001");
        paramMap.put("plate", "浙 A99999");
        paramMap.put("plateColor", "0");
        paramMap.put("createTime", "20201030124512");
        paramMap.put("startTime", "20201101000000");
        paramMap.put("endTime", "20201130235959");
        paramMap.put("userName", "李三");
        paramMap.put("userPhone", "18800002222");

        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "addVehicle");
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("三方 URL", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":1}

3.4.2 删除会员车【v3.0】(下行)

删除会员车(delVehicle)	
概述：	删除停车场会员车信息；

接口请求参数

字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
plate	String	32	是	车牌号码

请求示例代码：

```
private static void sendDelVehicle() throws Exception{
    String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
    String dataKey = "91305fbf367df8e4";
    String signKey = "2c1743a391305fbf367df8e4f069f9f9";
    String account = "alpha";

    Map<String, Object> paramMap = new HashMap<>();
    paramMap.put("parklotCode", "100001");
    paramMap.put("plate", "浙 A99999");

    Map<String, String> contentMap = new HashMap<String, String>();
    contentMap.put("method", "delVehicle");
    contentMap.put("param", GsonUtil.toJson(paramMap));

    String content = GsonUtil.toJson(contentMap);

    Map<String, String> reqMap = new HashMap<String, String>();
    reqMap.put("account", account);
    reqMap.put("timeStamp", timeStamp);
    reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
    reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

    String json = HttpPost.sendJsonObject("三方 URL", reqMap);

    System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":1 }

3.4.3 查询会员车【v3.0】(下行)

查询会员车(queryVehicle)				
概述:	查询停车场会员车信息			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
plate	String	32	是	车牌号码

请求示例代码:

```
private static void sendQueryVehicle() throws Exception{
        String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
        String dataKey = "91305fbf367df8e4";
        String signKey = "2c1743a391305fbf367df8e4f069f9f9";
        String account = "alpha";

        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("parklotCode", "100001");
        paramMap.put("plate", "浙 A99999");

        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "queryVehicle");
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("三方 URL", reqMap);

        System.out.println(json);
}
```

请求响应结果				
字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			查询结果数组对象

+plate	String	32	是	车牌号码
+plateColor	String	32	是	车牌颜色编号
+createTime	String	32	是	创建时间，yyyyMMddHHmmss
+startTime	String	32	是	开始时间，yyyyMMddHHmmss
+endTime	String	32	是	结束时间，yyyyMMddHHmmss
+userName	String	32	否	车主姓名
+userPhone	String	32	否	车主联系方式

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":[{"plate":"浙A99999","plateColor":"0","createTime":"20201030124512","startTime":"20201101000000","endTime":"20201130235959","userName":"李三","userPhone":"18800002222"}]}

3.5. 收费员

3.5.1 添加收费员【v3.0】(下行)

添加收费员（addParklotUser）				
概述：	停车场添加收费员账号			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
userCode	String	32	是	收费员账号
passWd	String	32	是	收费员账号对应的密码，MD5 加密字符串
enable	int	2	是	是否启用，0-不启用，1-启用

请求示例代码：

```
private static void sendAddParklotUser() throws Exception{
    String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
    String dataKey = "91305fbf367df8e4";
    String signKey = "2c1743a391305fbf367df8e4f069f9f9";
    String account = "alpha";

    Map<String, Object> paramMap = new HashMap<>();
    paramMap.put("parklotCode", "100001");
    paramMap.put("userCode", "001");
    paramMap.put("passWd", "e10adc3949ba59abbe56e057f20f883e");
    paramMap.put("enable", 1);

    Map<String, String> contentMap = new HashMap<String, String>();
    contentMap.put("method", "addParklotUser");
```

```
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("三方 URL", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":null }

3.5.2 修改收费员【v3.0】(下行)

修改收费员（updateParklotUser）	
概述：	停车场修改收费员账号

接口请求参数

字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
userCode	String	32	是	收费员账号
passWd	String	32	是	收费员账号对应的密码，MD5 加密字符串
enable	int	2	是	是否启用，0-不启用，1-启用

请求示例代码：

```
private static void sendUpdateParklotUser() throws Exception{
        String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
        String dataKey = "91305fbf367df8e4";
        String signKey = "2c1743a391305fbf367df8e4f069f9f9";
        String account = "alpha";

        Map<String, Object> paramMap = new HashMap<>();
```

```
        paramMap.put("parklotCode", "100001");
        paramMap.put("userCode", "001");
        paramMap.put("passWd", "e10adc3949ba59abbe56e057f20f883e");
        paramMap.put("enable", 0);

        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "updateParklotUser");
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("三方 URL", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":null }

3.5.3 删除收费员【v3.0】(下行)

删除收费员(delParklotUser)				
概述：	删除停车场收费员账号；			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
userCode	String	32	是	收费员账号

请求示例代码：

```
private static void senddelParklotUser() throws Exception{
        String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
```

```
        String dataKey = "91305fbf367df8e4";
        String signKey = "2c1743a391305fbf367df8e4f069f9f9";
        String account = "alpha";

        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("parklotCode", "100001");
        paramMap.put("userCode", "001");

        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "delParklotUser");
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("三方 URL", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":null,"httpStatus":200}

3.5.4 获取停车场所有收费员【v3.0】(下行)

获取停车场所有收费员(queryParklotUserList)				
概述：	获取停车场所有收费员账号信息			

接口请求参数

字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号

请求示例代码：

```
private static void sendGetParklotUser() throws Exception{
```

```
        String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
        String dataKey = "91305fbf367df8e4";
        String signKey = "2c1743a391305fbf367df8e4f069f9f9";
        String account = "alpha";

        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("parklotCode", "100001");

        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "queryParklotUserList");
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("三方 URL", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			查询结果数组对象
+userCode	String	32	是	收费员账号
+passWd	String	32	是	收费员账号对应的密码，MD5 加密字符串
+enable	int	2	是	是否启用，0-不启用，1-启用

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":[{"userCode":"15355071025","passWd":"e10adc3949ba59abbe56e057f20f883e","enable":1},{"userCode":"15857166793","passWd":"e10adc3949ba59abbe56e057f20f883e","enable":1},{"userCode":"1001","passWd":"98134FB4E18DE4DD42AD1CD8FBA5FB4B","enable":1},{"userCode":"1002","passWd":"E10ADC3949BA59ABBE56E057F20F883E","enable":1},{"userCode":"1005","passWd":"E10ADC3949BA59ABBE56E057F20F883E","enable":1},{"userCode":"001","passWd":"e10adc3949ba59abbe56e057f20f883e","enable":0}],"httpStatus":200}

3.5.5 收费员登录/登出记录【v3.0】(上行)

收费员登录/登出记录(parklotUserLoginout)				
概述:	实时上报收费员登录/登出记录信息;			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
userCode	String	32	是	收费员账号
type	int	2	是	操作类型，0-登录，1-登出
time	String	32	是	时间，yyyyMMddHHmmss

请求示例代码:

```
private static void sendParklotUserLoginout() throws Exception{
    String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
    String dataKey = "91305fbf367df8e4";
    String signKey = "2c1743a391305fbf367df8e4f069f9f9";
    String account = "alpha";

    Map<String, Object> paramMap = new HashMap<>();
    paramMap.put("parklotCode", "100001");
    paramMap.put("userCode", "001");
    paramMap.put("type", 0);
    paramMap.put("time", DateUtil.format(new Date(), "yyyyMMddHHmmss"));

    Map<String, String> contentMap = new HashMap<String, String>();
    contentMap.put("method", "parklotUserLoginout");
    contentMap.put("param", GsonUtil.toJson(paramMap));

    String content = GsonUtil.toJson(contentMap);

    Map<String, String> reqMap = new HashMap<String, String>();
    reqMap.put("account", account);
    reqMap.put("timeStamp", timeStamp);
    reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
    reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

    String json = HttpPost.sendJsonObject("http://test/openapi/v3/dclot", reqMap);

    System.out.println(json);
}
```

请求响应结果				
字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000

msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","httpStatus":200,"data":1}

3.6. 黑名单

3.6.1 添加/修改黑名单【v3.0】(下行)

添加/修改黑名单(addBlackVehicle)				
概述：	若停车场已经存在此车则直接更新，若不存在做新增处理；			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
plate	String	32	是	车牌号码
startTime	String	32	是	开始时间，yyyyMMddHHmmss
endTime	String	32	是	结束时间，yyyyMMddHHmmss
limitPassIn	int	2	是	是否限制入场，0-不限制，1-限制
vehContact	String	32	否	车辆联系人
vehContactNumber	String	32	否	车辆联系人电话
reason	String	512	是	拉黑原因

请求示例代码：

```
private static void sendAddBlackVehicle() throws Exception{
    String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
    String dataKey = "91305fbf367df8e4";
    String signKey = "2c1743a391305fbf367df8e4f069f9f9";
    String account = "alpha";

    Map<String, Object> paramMap = new HashMap<>();
    paramMap.put("parklotCode", "100001");
    paramMap.put("plate", "浙 A99999");
    paramMap.put("limitPassIn", 1);
    paramMap.put("startTime", "20201101000000");
    paramMap.put("endTime", "20201130235959");
    paramMap.put("vehContact", null);
    paramMap.put("vehContactNumber", null);
    paramMap.put("reason", "道路欠费 500 元");
```

```
        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "addBlackVehicle");
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("三方 URL", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":1,"httpStatus":200}

3.6.2 删除黑名单【v3.0】(下行)

删除黑名单(delBlackVehicle)	
概述：	删除停车场黑名单车辆；

接口请求参数

字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
plate	String	32	是	车牌号码

请求示例代码：

```
private static void sendDelBlackVehicle() throws Exception{
        String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
        String dataKey = "91305fbf367df8e4";
        String signKey = "2c1743a391305fbf367df8e4f069f9f9";
        String account = "alpha";

        Map<String, Object> paramMap = new HashMap<>();
```

```
        paramMap.put("parklotCode", "100001");
        paramMap.put("plate", "浙 A99999");

        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "delBlackVehicle");
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("三方 URL", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":1,"httpStatus":200}

3.7. 道闸

3.7.1 开闸【v3.0】(下行)

开闸(openLaneDevice)				
概述：	运营系统远程指定停车场开闸；			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
laneCode	String	32	是	停车场的车道编号

请求示例代码：

```
private static void sendOpenLane() throws Exception{
```

```
        String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
        String dataKey = "91305fbf367df8e4";
        String signKey = "2c1743a391305fbf367df8e4f069f9f9";
        String account = "alpha";

        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("parklotCode", "100001");
        paramMap.put("laneCode", "415F4E69D1648E580065_1");

        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "openLaneDevice");
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("三方 URL", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":1,"httpStatus":200}

3.7.2 锁闸【v3.0】(下行)

锁闸(lockLaneDevice)				
概述：	运营系统远程指定停车场锁闸；			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
laneCode	String	32	是	停车场的车道编号

请求示例代码：

```
private static void sendLockLane() throws Exception{
        String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
        String dataKey = "91305fbf367df8e4";
        String signKey = "2c1743a391305fbf367df8e4f069f9f9";
        String account = "alpha";

        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("parklotCode", "100001");
        paramMap.put("laneCode", "415F4E69D1648E580065_1");

        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "lockLaneDevice");
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("三方 URL", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":1,"httpStatus":200}

3.7.3 解锁【v3.0】(下行)

解锁(unlockLaneDevice)	
概述：	运营系统远程指定停车场解锁道闸;

接口请求参数

字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号

laneCode	String	32	是	停车场的车道编号

请求示例代码：

```
private static void sendUnLockLane() throws Exception{
    String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
    String dataKey = "91305fbf367df8e4";
    String signKey = "2c1743a391305fbf367df8e4f069f9f9";
    String account = "alpha";

    Map<String, Object> paramMap = new HashMap<>();
    paramMap.put("parklotCode", "100001");
    paramMap.put("laneCode", "415F4E69D1648E580065_1");

    Map<String, String> contentMap = new HashMap<String, String>();
    contentMap.put("method", "unlockLaneDevice");
    contentMap.put("param", GsonUtil.toJson(paramMap));

    String content = GsonUtil.toJson(contentMap);

    Map<String, String> reqMap = new HashMap<String, String>();
    reqMap.put("account", account);
    reqMap.put("timeStamp", timeStamp);
    reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
    reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

    String json = HttpPost.sendJsonObject("三方 URL", reqMap);

    System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":1,"httpStatus":200}

3.8. 监控

3.8.1 停车场设备状态【v3.0】(上行)

停车场设备状态(deviceStatus)				
概述:	停车场定时上报停车场设备运营状态信息			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
laneCode	String	32	是	停车场的车道编号
cameraStatus	int	2	是	车道所在相机在线状态，0-离线，1-在线

请求示例代码:

```
private static void sendDeviceStatus() throws Exception{
        String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
        String dataKey = "91305fbf367df8e4";
        String signKey = "2c1743a391305fbf367df8e4f069f9f9";
        String account = "alpha";

        List<Object> deviceList = new ArrayList<>();
        Map<String, Object> dev1 = new HashMap<>();
        dev1.put("parklotCode", "100001");
        dev1.put("laneCode", "415F4E69D1648E580065_1");
        dev1.put("cameraStatus", 1);
        Map<String, Object> dev2 = new HashMap<>();
        dev2.put("parklotCode", "100001");
        dev2.put("laneCode", "415F4E69D1648E580065_2");
        dev2.put("cameraStatus", 0);
        deviceList.add(dev1);
        deviceList.add(dev2);

        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "deviceStatus");
        contentMap.put("param", GsonUtil.toJson(deviceList));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("http://test/openapi/v3/dclot", reqMap);

        System.out.println(json);
}
```

请求响应结果				
字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，成功 000000
msg	String	128	是	结果说明
data	Object			NULL
响应结果示例： **{"code":"000000","msg":"请求服务正常返回","httpStatus":200,"data":1}**				

3.8.2 停车场心跳【v3.0】(下行)

停车场心跳(parklotHeart)				
概述：	运营系统定时 5 分钟向三方系统发送请求，目的是检测下行线路是否正常；同时是指定停车场进行巡检；			
接口请求参数				
字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
uploadDataUrl	String	256	否	三方系统上报数据的地址，此字段一般用于运营系统更换了地址后下行告知三方，三方自动调整上行地址

请求示例代码：
`private static void sendParklotHeart() throws Exception{`
`String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");`
`String dataKey = "91305fbf367df8e4";`
`String signKey = "2c1743a391305fbf367df8e4f069f9f9";`
`String account = "alpha";`
 	
`Map<String, Object> paramMap = new HashMap<>();`
`paramMap.put("parklotCode", "100001");`
`paramMap.put("uploadDataUrl", "http://test/openapi/v3/dclot");`
`// paramMap.put("uploadDataUrl", null);`
 	
`Map<String, String> contentMap = new HashMap<String, String>();`
`contentMap.put("method", "parklotHeart");`
`contentMap.put("param", GsonUtil.toJson(paramMap));`
 	
`String content = GsonUtil.toJson(contentMap);`
 	
`Map<String, String> reqMap = new HashMap<String, String>();`
`reqMap.put("account", account);`				

```
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("三方 URL", reqMap);

        System.out.println(json);
    }
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，正常 000000，其他编号或超时表示非正常
msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":1,"httpStatus":200}

3.8.3 停车场空余车位【v3.1】(上行)

停车场空余车位(parklotEmptyBerth)	
概述：	停车场在车辆入场或离场后实时上报运营系统最新空余信息；

接口请求参数

字段名	字段类型	长度	是否必填	描述
parklotCode	String	32	是	运营系统提供的车场编号
totalBerthNum	int	11	是	停车场总泊位数
emptyBerthNum	int	11	是	停车场当前空余泊位数
regionId	int	11	否	内部场区区域 ID，默认为 1，可不填

请求示例代码：

```
private static void sendParklotEmptyBerth() throws Exception{
    String timeStamp = DateUtil.format(new Date(), "yyyyMMddHHmmss");
    String dataKey = "91305fbf367df8e4";
    String signKey = "2c1743a391305fbf367df8e4f069f9f9";
    String account = "alpha";

    Map<String, Object> paramMap = new HashMap<>();
    paramMap.put("parklotCode", "100001");
    paramMap.put("totalBerthNum", 60);
    paramMap.put("emptyBerthNum", 35);
    paramMap.put("regionId", 1);
```

```
        Map<String, String> contentMap = new HashMap<String, String>();
        contentMap.put("method", "parklotEmptyBerth");
        contentMap.put("param", GsonUtil.toJson(paramMap));

        String content = GsonUtil.toJson(contentMap);

        Map<String, String> reqMap = new HashMap<String, String>();
        reqMap.put("account", account);
        reqMap.put("timeStamp", timeStamp);
        reqMap.put("sign", SignUtil.createSign(account, timeStamp, signKey));
        reqMap.put("content", AesUtil.encryptCBC(content, dataKey));

        String json = HttpPost.sendJsonObject("三方 URL", reqMap);

        System.out.println(json);
}
```

请求响应结果

字段名	字段类型	长度	是否必填	描述
code	String	32	是	返回码，正常 000000，其他编号或超时表示非正常
msg	String	128	是	结果说明
data	Object			NULL

响应结果示例：

{"code":"000000","msg":"请求服务正常返回","data":1,"httpStatus":200}

4. 针对车场数据上报接入说明

如果只上报数据到城管监管平台，需要接入的接口如下：

3.8.3 停车场空余车位

5. 附件

4.1 返回码

返回码	返回说明
000000	请求服务正常返回
010201	无效账号
010202	无效方法 method
010203	数据转换异常
010204	时间戳格式错误
010205	超时请求，时间戳与服务收到请求时间差超过 N 分钟，默认 10 分钟
010206	签名校验失败
010207	数据解密失败
010208	解密成功，数据解析失败，即解析出来的数据转换成对象失败
010209	此功能未开放
010210	未知支付方式
900400	请求参数不正确
900404	请求名称不存在
900405	HTTP 请求方法不正确
900500	HTTP 处理请求异常
999999	恶意攻击

4.2 车道类型

1	入口
2	出口-不收费
3	出口-收费

4.3 车牌颜色

0	蓝色（默认）
1	黄色
2	黑色
3	白色
4	绿色
5	黄绿色

4.4 车辆类型

0	其他
1	小型车
2	大型车

4.5 放行方式

50	牌识放行
51	刷卡放行
52	取卡放行
53	收卡放行
54	手动放行
55	异常放行（离场时未匹配到入场记录、离场时产生了费用但是免费放行）
56	禁止通行
57	平台放行（由运营系统控制的放行）
58	遥控放行
59	未知方式

4.6 支付方式

0	现金
1	微信
2	支付宝
3	市民卡
4	ETC
5	银联
6	优惠券
7	运营系统离场缴费
8	运营系统预缴费

4.7 支付模式

1	离场缴费，即全额缴费，查询停车费是多少就缴费多少，停车场可校验预付账单对应的金额是否一致
2	预缴费，运营平台给车辆提供优惠，提前给车辆交部分停车费，剩余金额需要车主自行缴费；停车场不可校验预付账单金额是否一致； 如查询停车费是 10 元，运营系统提前给车主支付 5 元，余下 5 元车主自行缴费