

位位通-停车场系统接入文档

2021.8.4

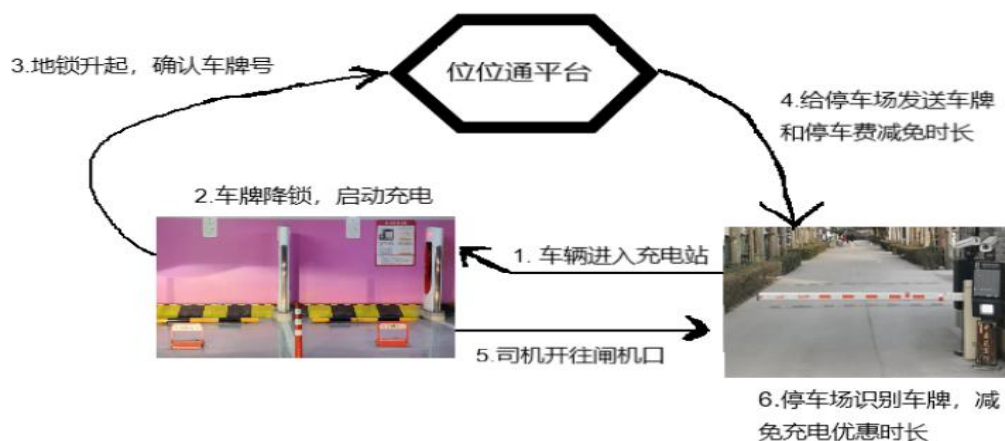
1. 背景说明

1.1 适用对象

本文档描述了位位通相机识别降锁业务场景下，位位通平台与停车场系统对接说明，适用于停车场系统的产品与技术同学。

1.2 业务场景

业务场景如下图所示：



1. 充电司机驾车通过闸机进入充电站；
2. 车位上方的相机识别绿牌降锁，司机进行停车充电；
3. 司机驶离地锁升起，现场网关向位位通平台回传充电订单完成、车牌号等信息；
4. 位位通平台向停车场系统发送停车减免信息，包括车牌号、减免时长等信息；
5. 司机离开充电站，开往停车场出口道闸；
6. 停车场系统识别车牌，减免充电优惠时长，根据规则收取剩余费用；

注意：

1. 停车场系统与位位通平台之间的交互仅有一个充电减免接口；
2. 车辆一次入场与离场之间，最多减免 120 分钟，如果在这期间停车场系统多次收到减免信息，需要停车场系统做去重逻辑，避免叠加减免，造成停车场资损；
3. 停车场系统必须提供一个**固定的外网可访问的域名或公网 IP 地址**；

4. 如果停车系统有平台，**位位通平台优先与停车系统平台对接**，这样便于接入停车场系统平台下的其他场站，同时便于位位通访问停车系统平台的固定域名；

2. 接入说明

2.1 充电停车减免接口说明

位位通平台采用 HTTP POST 方式调用停车场系统，停车场系统根据以下报文格式 供对应的 HTTP 服务即可。

注意：

1. 该接口必须是一个外网可访问的接口，即该接口方必须提供一个固定的外网可访问的域名或公网 IP 地址

2.1.1 HTTP 报文头

HTTP Headers Key	HTTP Headers Value
Content-type	application/json; charset=UTF-8

说明：报文体将采用 json 格式。

2.1.2 HTTP 报文体

1) 请求参数

参数	类型	描述
plateNo	string	车牌号
merchId	string	停车场唯一标识（由停车场系统提供）
duration	string	减免时长（单位：分钟）
sign	string	签名（调用方按照签名规则生成，停车场系统通过眼前规则校验签名是否正确；签名密钥由位位通充电平台颁发）

参数示例：

```
{
  "duration": "40",
  "plateNo": "京 XJ1236",
  "sign": "0DD44E081F0375C59FDA6ED4946817E8",
  "merchId": "1"
}
```

2) 响应参数

参数	类型	描述
code	int	10000: 减免成功, 其他 code 可自定义, 例如{"code":20033, "msg":"减免失败, 车辆不在场内", "data": null}
msg	string	code 对应的描述
data	int	其他响应数据, 留作扩展, 没有使用时填 null

参数示例:

```
{
  "code": 20002,
  "msg": "车辆不在场内",
  "data": {}
}
```

2.2 签名规则

请求参数中 sign 值由签名规则 genSign (签名规则逻辑可查看以下代码样例) 生成, signKey 由位位通平台颁发。

以下是代码样例:

```
public void test(String plateNo, String merchId, String duration) {
    // 先将获取的三个参数放入一个 map 集合
    Map<String, String> paramMap = Maps.newHashMap();
    paramMap.put("plateNo", plateNo);
    paramMap.put("merchId", merchId);
    paramMap.put("duration", duration);
    // 调用 genSign 方法, 将组装好的 map 集合和签名秘钥作为参数传入
    genSign(paramMap, signKey);
}

public static String genSign(final Map<String, String> paramMap, String signKey) {
    // 创建一个长度和 map.size()大小一致的字符串数组
    String[] keyArray = new String[paramMap.keySet().size()];
    // 将 map 的 key 的 set 集合转化为数组
    paramMap.keySet().toArray(keyArray);
    // 将数组元素按字典序进行排序
    Arrays.sort(keyArray);
    StringBuffer sb = new StringBuffer();
    // 遍历 key 数组
    for (String key : keyArray) {
        // 将键与值按照 key=value&进行拼接
        if (StringUtils.isNotBlank(key) && StringUtils.isNotBlank(paramMap.get(key)))
            { sb.append(key).append("=").append(paramMap.get(key)).append("&");
            }
    }
}
```

```

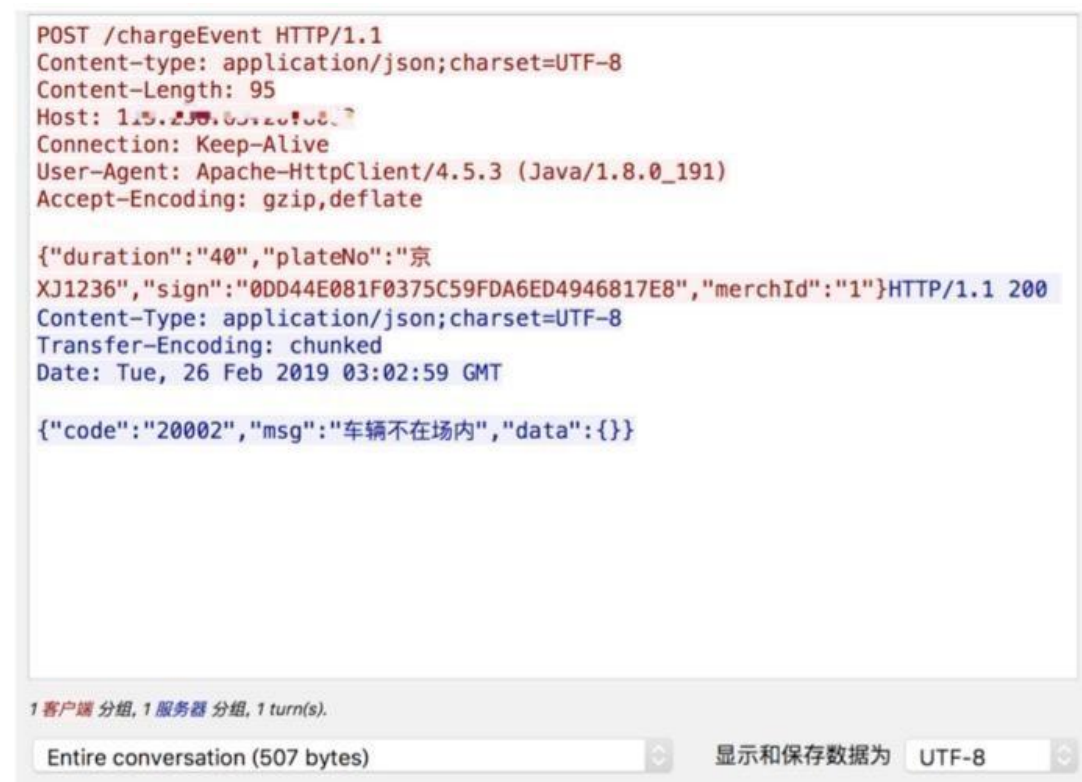
}
// 拼接秘钥时先将秘钥做一次 md5 加密然后拼接
sb.append("key=").append(DigestUtils.md5Hex(signKey));
logger.info("param : {}", sb.toString());
// 将拼接的字符串先进行 md5 加密，再转为大写
String sign = DigestUtils.md5Hex(sb.toString()).toUpperCase();
logger.info("param: {} sign: {}", sb.toString(), sign);
return sign;
}

```

整体逻辑：

1. 将前三个参数的参数名按 ASCII 码从小到大（A~Z）排序（字典序）
2. 按 参数名 1=参数值 1&参数名 2=参数值 2 进行拼接
3. 最后拼接秘钥时，先将秘钥做一次 md5 加密转小写再拼接
4. 将拼接的字符串先进行 md5 加密，再转为大写
5. 生成签名

2.3 报文样例

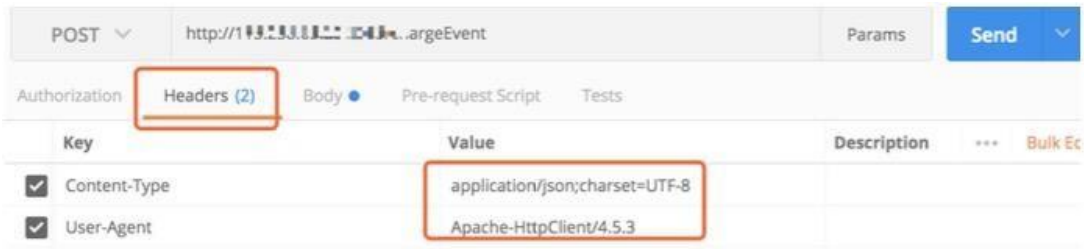


2.4 开发自测

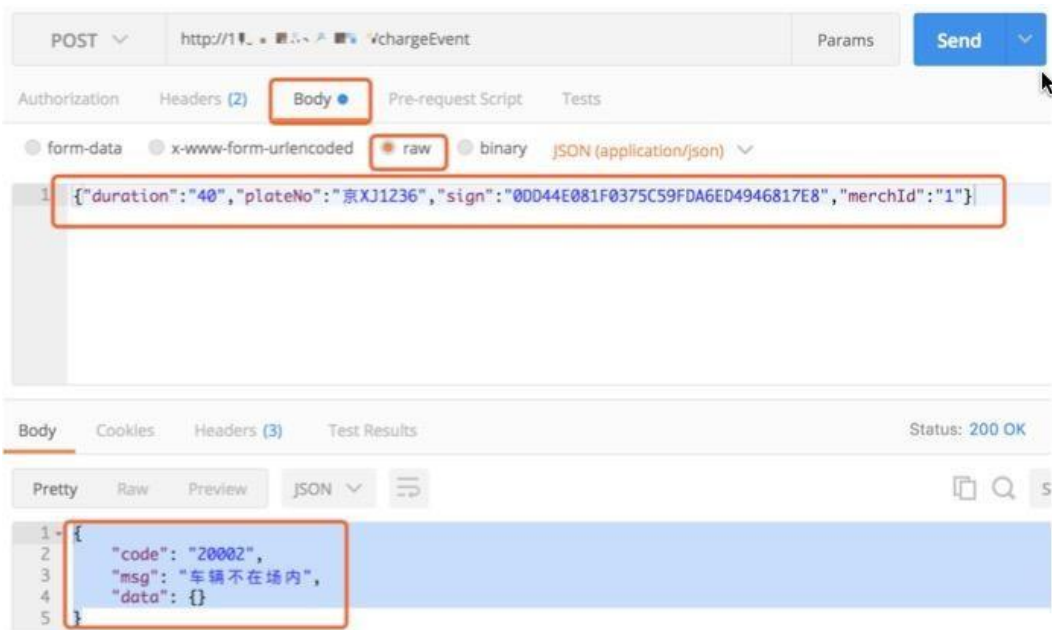
在开发自测阶段，停车场系统的开发同学可以使用 postman 工具（下载地址：

<https://www.getpostman.com/downloads/>) 来模拟位位通充电平台向停车场系统发起调用，配置样例如下：

Header 配置：



Body 配置：



Body 样例：

```
{"duration":"40","plateNo":"XJ1236","sign":"0DD44E081F0375C59FDA6ED4946817E8","merchId":"1"}
```

2.5 现场实车测试

停车场系统完成上述充电停车减免接口的开发、自测、上线后，请联系位位通充电的业务接口人安排现场实车测试。实车测试情景包括：

测试情景	预期
司机完成充电，且停车时长小于位位通推送的减免时长	司机不需支付停车费
司机完成充电，且停车时长大于位位通推送的减免时长	司机需支付超出位位通减免时长的停车费

完成实车测试后，系统开始线上运营。