中间件协议文档

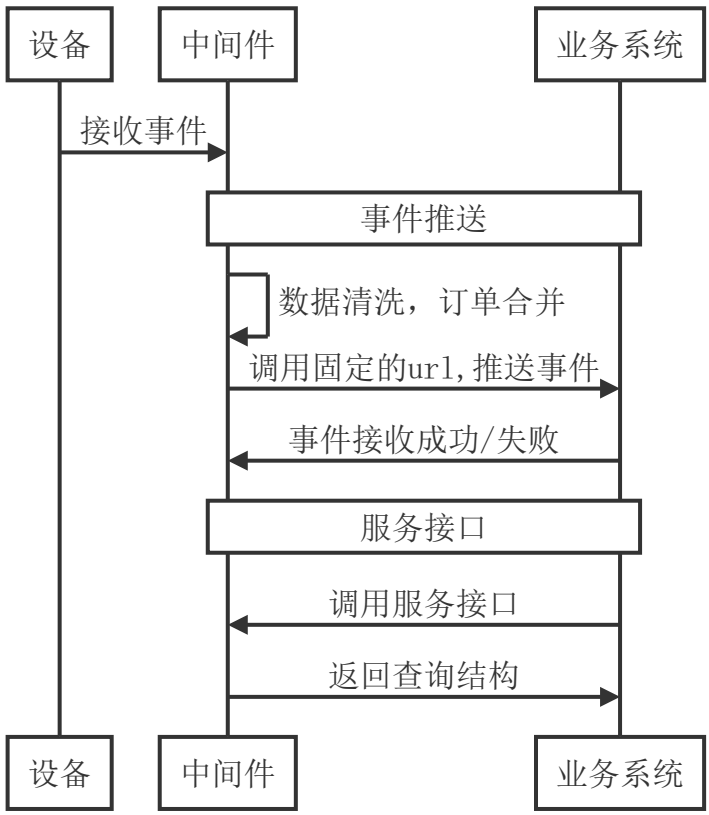
1.文档说明

本文针对人群	技术人员
内容说明	停车中间件提供的【事件推送】和【服务接口】
注意事项	根据业务系统需求不同，只需使用部分事件和接口即可，请按需使用

2.修订记录

版本号	修订人	修订日期	修订描述
V1.0	何旭	2020-11-02	初稿
V1.1	何旭	2020-12-09	事件推送字段内容修改，适应客户的不同需求

3.业务流程



事件：泛指停车事件，如：入场、出场.....等。

中间件：停车事件中转处理系统，包括订单合并，向业务系统推送事件，提供服务接口等功能。

事件推送：业务系统根据本文档定义，提供一个固定的“事件接收接口”，中间件系统通过调用该接口实现事件推送

服务接口：指业务系统调用中间件的查询接口以获得服务。

4.全局定义

协议：

本文档所涉及的接口定义均采用http协议

数据格式：

数据格式遵守json数据格式

调用服务接口规范：

请求参数封装为json数据格式后，必须将参数放入请求体（body）。

请求头必须附带Content-Type = application/json。

中间件将以json格式返回结果。

事件接收接口规范：

中间件将按照json规范，从请求体（body）中推送事件。

请求头将附带Content-Type = application/json。

接口响应码定义：

errorcode	响应码说明	类型
0	成功	Int
1	失败	Int
4	参数错误	Int

事件类型定义：

evt	响应码说明	类型
1	入场	Int
2	停稳	Int
4	出场	Int
8	空场	Int

5.事件推送

人工审核通过的订单将推送到客户的SAAS平台，需要客户提供：

业务系统接收地址：http://{ip}:{port}/xxx/xxx/xxxx

入参：

参数名	描述	类型
orderId	所属订单Id	String
parkingName	车场名称	String
parkingCode	车场编号	Int
deviceSn	设备序列号	String
berthCode	泊位号	String
plateNumber	车牌号	String
plateColor	车牌颜色：0:未知; 1:蓝色; 2:黄色; 3:白色; 4:黑色; 5:绿色;	Int
timeIn	入场时间 (单位：秒)	Int
timeOut	出场时间（单位：秒）	int

响应：

参数	描述	类型
errorcode	错误码，参见接口响应码定义	String
message	响应描述	String

示例：

```
入参json：
{
    "orderId": "20200927104355355774334",
    "parkingName": "天府三街中段",
    "parkingCode": 35,
    "berthCode":"SJ-0001",
    "deviceSn":"7badde28-179f5917",
    "plateNumber": "粤D18436",
    "plateColor": 1,
    "timeIn":1603023395,
    "timeOut":1603024390,
}
响应json：
{
    "errorcode":0,
    "message": ""
}
```

6.服务接口

中间件的查询接口都需要进行签名认证，请参考签名认证

设备查询接口：

描述：通过设备序列号查询设备相关信息

PATH	方法
/v2/server_api/query/device	GET

入参:

参数	描述	类型
deviceSn	设备序列号	String

响应:

参数	描述	类型
deviceSn	设备序列号	String
parkingCode	车场编号	Int
parkingName	车场名称	String
deviceType	设备名称	String
deviceStatus	设备状态：0:在线 ,1:离线	Int
berths	泊位列表	list

示例:

```
请求 json
{
 "deviceSn": "7badde28-179f5917"
}
响应 json
//成功
{
 "errorcode": "0",
 "message": "",
 "data":[
     {
         "deviceSn": "7badde28-179f5917",
         "deviceType": "H1L",
         "deviceStatus": 0,
         "parkingCode": "400",
         "parkingName": "天府三街中段",
         "berths":[
             "SJ-001",
             "SJ-002",
             "SJ-003",
             "SJ-004"
         ]
     }
  ]
}
//失败
{
 "errorcode": "1",
```

```
 "message": "Request failed!",
}
```

泊位查询接口：

描述：通过车场编号查询该车场所有泊位信息

PATH	方法
/v2/server_api/query/berths	GET

入参：

参数	描述	类型
parkingCode	车场编号	Int

响应：

参数	描述	类型
parkingCode	车场编号	String
parkingName	车场名称	String
deviceSn	设备序列号	String
berthCode	泊位号	String
berthState	泊位状态(0:空闲，1：使用中)	Int
plateNumber	车牌号	String
plateColor	车牌颜色（0:未知; 1:蓝色; 2:黄色; 3:白色; 4:黑色; 5:绿色）	Int
timeIn	驶入时间（单位：秒）	Int

示例：

```
请求 json
{
 "parkingCode": "400",
}
响应 json
//成功
{
 "errorcode": "0",
 "message": "",
 "data":[
     {
         "parkingCode": "400",
         "parkingName": "天府三街中段",
         "deviceSn": "7badde28-179f5917",
         "berthCode":"SJ-0001",
         "berthState":1,
```

```
            "plateNumber":"苏ABU082",
            "plateColor":1,
            "timeIn":1603023378
        },
        {
            "parkingCode": "400",
            "parkingName": "天府三街中段",
            "deviceSn": "e392770f-bf414754",
            "berthCode":"SJ-0002",
            "berthState":0,
            "plateNumber":"",
            "plateColor":0,
            "timeIn":0
        },
    ]
}
//失败
{
 "errorcode": "1",
 "message": "Request failed!",
}
```

车场查询接口：

描述：通过车场编号，获得车场基本信息

PATH	方法
/v2/server_api/query/parking	GET

入参：

参数	描述	类型
parkingCode	车场编号	Int

响应：

参数	描述	类型
parkingCode	车场编号	Int
parkingName	车场名称	String
deviceSns	设备序列号列表	list

示例：

```
请求 json
{
 "parkingCode": "400",
}
响应 json
//成功
```

```
{
 "errorcode": "0",
 "message": "",
 "data":[
     {
         "parkingCode": "400",
         "parkingName": "天府三街中段",
         "deviceSns":[
             "f2d79c79-10907885":
             "f2d79c79-10907885",
             "f2d79c79-10907885",
             "f2d79c79-10907885",
             "f2d79c79-10907885",
         ]
     }
  ]
}
//失败
{
 "errorcode": "1",
 "message": "Request failed!",
}
```

订单查询接口：

描述：调用此接口，可以查询某车场在一段时间内产生的订单，也可以指定查询车场中具体某台设备在一段时间产生的订单

PATH	方法
/v2/server_api/query/orders	GET

入参：

参数	描述	类型
deviceSn	设备序列号（该字段用于查询车场指定设备产生的订单）	String
parkingCode	车场编号	Int
startTime	起始时间，单位：秒	Int
endTime	结束时间，单位：秒	Int

响应：

参数	描述	类型
parkingCode	车场编号	Int
parkingName	车场名称	String
orders	订单列表	list
id	订单编号	String
deviceSn	设备序列号	String
berthCode	泊位号	String
plateNumber	车牌号	String
plateColor	车牌颜色	Int
timeIn	驶入时间（单位：秒）	Int
timeOut	驶出时间（单位：秒）,如果为0，代表车辆还未驶出	Int

示例：

```
请求 json
{
 "deviceSn": "",
 "parkingCode": 301,
 "startTime":1603023378,
 "endTime":1603024456,
}
响应 json
//成功
{
 "errorcode": "0",
 "message": "",
 "data":[
     {
         "parkingCode": 301,
         "parkingName": "天府三街中段",
         "orders":[
             {
                 "id":"2020092710435535577433 4",
                 "deviceSn":"473cced4-4e1074c1",
                 "berthCode":"SJ-0005",
                 "plateNumber":"粤D18436",
                 "plateColor":1,
                 "timeIn":1603023395,
                 "timeOut":1603024390,
             },
              {
                 "id":"2020092710435535577433 4",
                 "deviceSn":"473cced4-4e1074c1",
                 "berthCode":"SJ-0005",
                 "plateNumber":"粤D18436",
                 "plateColor":1,
                 "timeIn":1603023395,
                 "timeOut":1603024390,
```

```
            },
              {
                "id":"20200927104355355774334",
                "berthCode":"SJ-0005",
                "plateNumber":"粤D18436",
                "plateColor":1,
                "timeIn":1603023395,
                "timeOut":1603024390,
            },
        ]
    }
  ]
}
//失败
{
 "errorcode": "1",
 "message": "Request failed!",
}
```

停车事件查询接口:

描述：通过订单号查询具体的停车事件

PATH	方法
/v2/server_api/query/order/detial	GET

入参:

参数	描述	类型
orderId	订单号	String

响应:

参数	描述	类型
orderId	订单号	
deviceSn	设备序列号	String
berthCode	泊位号	String
evt	事件类型,，参见事件类型定义	Int
plateNumber	车牌号	String
plateColor	车牌颜色（0:未知; 1:蓝色; 2:黄色; 3:白色; 4:黑色; 5:绿色）	Int
picUrl	图片路径	String
recordTime	识别时间（单位：秒）	Int

示例:

请求 json

```
{
  "orderId": "2020092710435535577433 4",
}
```

响应 json

```
//成功
{
 "errorcode": "0",
 "message": "",
 "data": [
     {
         "orderId": "2020092710435535577433 4",
         "deviceSn": "473cced4-4e1074c1",
         "records":[
             {
                 "berthCode":"SJ-0005",
                 "plateNumber":"粤D18436",
                 "evt":1,
                 "plateColor":1,
                 "picUrl":"imgs/7badde28-
179f5917/20200927/1607_148_0_0_full.jpg",
                 "recordTime":1603023395
             },
                {
                 "berthCode":"SJ-0005",
                 "plateNumber":"粤D18436",
                 "evt":2,
                 "plateColor":1,
                 "picUrl":"imgs/7badde28-
179f5917/20200927/1607_148_0_1_full.jpg",
                 "recordTime":1603024495
             },
                 {
                 "berthCode":"SJ-0005",
                 "plateNumber":"粤D18436",
                 "evt":4,
                 "plateColor":1,
                 "picUrl":"imgs/7badde28-
179f5917/20200927/1607_148_0_2_full.jpg",
                 "recordTime":1603025595
             }
         ]
     }
  ]
}
//失败
{
 "errorcode": "1",
 "message": "Request failed!",
}
```

图片下载接口：

描述：通过picUrl获取图片的url

PATH	方法
/v2/server_api/query/image	GET

入参：

参数	描述	类型
picUrl	图片路径	String

响应：

参数	描述	类型
url	图片链接	String

示例：

```
请求 json
{
  "picUrl": "imgs/7badde28-179f5917/20200927/1607_148_0_1_full.jpg",
}
响应 json
//成功
{
 "errorcode": "0",
 "message": "",
 "data":{
     "url":"http://qiniu-test.vzicar.com/imgs/7badde28-
179f5917/20200927/1607_148_0_1_full.jpg?
e=1603031694&token=EEKLdQd1QLumi3eMdzv_a-
gEWZ3mDC4pKcM9DHGw:ngULau_hmiw2p4L86JPFxN34V2c="
 }
}
//失败
{
 "errorcode": "1",
 "message": "Request failed!",
}
```

7.签名认证

获取AccessKey:

登录路内停车管理平台，选择左侧导航栏**系统管理—AccessKey管理**菜单，创建自己的AccessKeyId，AccessKeySecret，过期时间选择永久

计算signature：

```
signature = urlencode(base64(hmac-sha1(AccessKeySecret,
          Method + "\n"
          + CONTENT-MD5 + "\n"
          + CONTENT-TYPE + "\n"
          + EXPIRES + "\n"
          + Resource)))
```

- AccessKeySecret: 表示签名所需要的密钥。
- Method: 表示`HTTP`请求的方法，主要有`GET`、`POST`、`PUT`、`DELETE`等，注意必须是大写。
- \n: 表示换行符。
- CONTENT-MD5: 表示请求内容数据的`MD5`值，对消息内容（不包括头部）计算`MD5`值获得128比特位数字，对该数字进行`base64`编码而得到。该请求头可用于消息合法性的检查（消息内容是否与发送时一致），如"eB5eJF1ptWaXm4bijSPyxw=="，如果请求中没有数据，则这个值为空。详情请参见RFC2616 Content-MD5。
- CONTENT-TYPE: 表示 内容的类型，如"`application/json`"，如果请求中没有数据，则这个值为空，如果请求中携带了数据，则一定要携带这个值。
- EXPIRES: 签名过期时间。这个参数的值是一个UNIX时间（自UTC时间1970年1月1号开始的秒数），用于标识该URL的超时时间。如果服务器接收到这个URL请求的时候晚于签名中包含的`expires`参数时，则返回请求超时的错误。为了安全起见，请尽可能设置一个较短的过期时间，比如10分钟。**注意：请确保您的计算机已同步正确的北京时间。**。
- Resource: 表示用户想要访问的资源。

其中有几点规则需要注意：

- 签名的字符串必须为`UTF-8`格式。含有中文字符的签名字符串必须先进行`UTF-8`编码，再与`AccessKeySecret`计算最终签名。
- 签名的方法用RFC 2104中定义的`HMAC-SHA1`方法，其中`Key`为`AccessKeySecret`。
- 服务器先验证请求时间是否晚于expires时间，然后再验证签名。
- 将签名字符串放到`URL`时，注意要对`URL`进行`urlencode`。

完整示例：

0. 前置条件

- 本示例中，约定AccessKey如下：
  - AccessKeyId: `7e9peQ8C1125A7Cz4LVFJl61jxFtHs0F`
  - AccessKeySecret: `ZfATtI0jK9uclIEwcHJ7JLAj7rRX1mgY`
- 本示例中，约定`expires`为：`1600689938`
- 本示例中的`Body Data`采用紧凑型的`Json`格式。

1. 计算CONTENT-MD5

- `Body Data`为`{"deviceSn":"6593d05e-34c85fbd"}`
- 按照上面`CONTENT-MD5`相关说明计算出结果为：`1ZGc8+QATeubmqayvGQ1fg==`

2. 计算CanonicalizedResource

针对以上示例，要访问的资源为：`v2/server_api/query/device`

3. 计算CanonicalString

`CanonicalString`为将要加密的字符串，格式为：

```
Method + "\n"
+ CONTENT-MD5 + "\n"
+ CONTENT-TYPE + "\n"
+ EXPIRES + "\n"
+ Resource
```

针对这个格式及以上计算出的结果，可以拼接出`CanonicalString`为：

```
GET\n1ZGc8+QATeubmqayvGQ1fg==\napplication/json\n1600689938\n/v2/server_api/query/device
```

注意：上面的`\n`是换行符，而不是字符串。

4. 计算signature

- 将`CanonicalString`转换为UTF-8编码，然后进行`HmacSHA1`加密，再进行`Base64`编码，得到结果为：`nafiCG84KDkF1hmmkA9ZgmG3CVk=`

5. 生成完成的URL

针对以上计算的结果，我们可以生成一个完整的`URL`，**注意针对每一个参数值要进行URL编码**，最终结果如下：

```
https://www.vzicar.com/v2/stp/user/devices/share?accesskey_id=7e9peQ8C1125A7Cz4LVFJl61jxFtHsOF&expires=1600689938&signature=nafiCG84KDkF1hmmkA9ZgmG3CVk%3D
```

调用设备查询接口示例：

```
func GetUrl (path string) string{
    accesskeyId := "7e9peQ8C1125A7Cz4LVFJl61jxFtHsOF"
    accesskeySecret := "ZfATtIOjK9uclIEwcHJ7JLAj7rRX1mgY"
    body := `{"deviceSn":"6593d05e-34c85fbd"}`
    expires := time.Now().Local().Unix() + 600
    resource := "/v2/server_api/query/device"
    contentType := "application/json"
    contentMD5 := getContentMD5(body)
    canonicalString := fmt.Sprintf("GET\n%s\n%s\n%d\n%s",contentMD5,contentType,expires, resource)
    signature := getBase64HmacSha1([]byte(accesskeySecret), []byte(canonicalString))
    signature = url.QueryEscape(signature)
    url := fmt.Sprintf("https://www.vzicar.com/v2/server_api/query/device?app_id=1&accesskey_id=%s&expires=%d&signature=%s", accesskeyId, expires, signature)
    return url
}

func getContentMD5(reqBody string) string {
    body, err := json.Marshal(reqBody)
    if err != nil {
        fmt.Println("生成json字符串错误")
    }
    h := md5.New()
```

```
    h.Write(body)
    return base64.StdEncoding.EncodeToString(h.Sum(nil))
}

func getBase64HmacSha1(key []byte, p []byte) string {
    h := hmac.New(sha1.New, key)
    h.Write(p)
    return base64.StdEncoding.EncodeToString(h.Sum(nil))
}
```