支付宝

IoT公交SDK集成说明文档

文档修改历史

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 作者 | 内容提要 | 发布日期 |
| V0.1 | 有凡 | [C]文档初稿 | 20190304 |
| V0.2 | 有凡 | [A]增加参数配置描述 | 20190426 |
| V0.3 | 琥珀 | [A]新增重要业务的Demo示例  [A]新增常见问题说明  [M]补充部分业务场景 | 20190428 |
| V0.4 | 乂宝 | [M]修改alipay\_sdk\_config参数定义 | 20190430 |
| V0.5 | 琥珀 | [M]3.1.7.6新增item\_id必填项 | 20190508 |

\*变化状态：C――创建，A——增加，M——修改，D——删除

目录

1 SDK介绍说明 5

1.1 术语 5

1.2 集成内容 5

1.2.1 IoT公交地铁SDK 5

2 SDK安装说明 6

2.1 支付宝离线验证SDK 6

2.1.1 集成说明 6

2.1.1.1 环境说明 6

2.1.1.2 文件说明 6

2.1.1.3 集成时序 8

3 调用接口说明 8

3.1.1 离线验证初始化 8

3.1.1.1 业务场景概述 8

3.1.1.2 接口使用概述 9

3.1.1.3 业务执行前提 9

3.1.1.4 业务规则 9

3.1.1.5 函数接口 9

3.1.1.6 入参 9

3.1.1.7 返回值 10

3.1.1.8 样例代码 10

3.1.2 SDK反初始化 10

3.1.2.1 业务场景概述 10

3.1.2.2 接口使用概述 10

3.1.2.3 业务执行前提 10

3.1.2.4 业务规则 11

3.1.2.5 函数接口 11

3.1.2.6 入参 11

3.1.2.7 返回值 11

3.1.2.8 样例代码 11

3.1.3 码信息解析 11

3.1.3.1 业务场景概述 11

3.1.3.2 接口使用概述 11

3.1.3.3 业务执行前提 11

3.1.3.4 业务规则 12

3.1.3.5 函数接口 12

3.1.3.6 入参 12

3.1.3.7 返回值 15

3.1.3.8 样例代码 15

3.1.4 码信息验证 15

3.1.4.1 业务场景概述 15

3.1.4.2 接口使用概述 15

3.1.4.3 业务执行前提 15

3.1.4.4 业务规则 15

3.1.4.5 函数接口 16

3.1.4.6 入参 16

3.1.4.7 返回值 18

3.1.4.8 样例代码 18

错误码说明 19

3.1.5 IoT日志功能初始化 19

3.1.5.1 业务场景概述 19

3.1.5.2 接口使用概述 20

3.1.5.3 业务执行前提 20

3.1.5.4 业务规则 20

3.1.5.5 函数接口 20

3.1.5.6 入参 20

3.1.5.7 返回值 20

3.1.5.8 样例代码 20

3.1.6 IoT设备控制 21

3.1.6.1 业务场景概述 21

3.1.6.2 接口使用概述 21

3.1.6.3 业务执行前提 21

3.1.6.4 业务规则 21

3.1.6.5 函数接口 21

3.1.6.6 入参 21

3.1.6.7 返回值 22

3.1.6.8 样例代码 22

3.1.6.9 DEMO示例 22

3.1.7 业务信息配置 22

3.1.7.1 业务场景概述 22

3.1.7.2 接口使用概述 22

3.1.7.3 业务执行前提 22

3.1.7.4 业务规则 23

3.1.7.5 函数接口 23

3.1.7.6 入参 23

3.1.7.7 返回值 24

3.1.7.8 样例代码 24

3.1.7.9 DEMO示例 24

3.1.8 日志同步[可选] 24

3.1.8.1 业务场景概述 24

3.1.8.2 接口使用概述 24

3.1.8.3 业务执行前提 25

3.1.8.4 业务规则 25

3.1.8.5 函数接口 25

3.1.8.6 入参 25

3.1.8.7 返回值 25

3.1.8.8 样例代码 25

3.1.9 日志采集 26

3.1.9.1 业务场景概述 26

3.1.9.2 接口使用概述 26

3.1.9.3 业务执行前提 26

3.1.9.4 业务规则 26

3.1.9.5 函数接口 26

3.1.9.6 入参 27

3.1.9.7 返回值 27

3.1.9.8 样例代码 27

3.1.9.9 DEMO示例 27

3.1.10 日志采集内存释放 28

3.1.10.1 业务场景概述 28

3.1.10.2 接口使用概述 28

3.1.10.3 业务执行前提 28

3.1.10.4 业务规则 28

3.1.10.5 函数接口 28

3.1.10.6 入参 29

3.1.10.7 返回值 29

3.1.10.8 样例代码 29

4 常见问题说明 29

4.1 执行install.sh报错 29

4.2 日志写入后不获取会不会造成空间无限增长？ 29

4.3 日志查询接口多久调用一次合适？ 30

4.4 日志查询接口获取数据后，需要解析再上传服务器？ 30

4.5 是否每次验码之前初始化都需要执行初始化？ 30

4.6 请求获取二维码信息成功，但请求验证时出现段错误，是什么原因？ 30

4.7 生成的脱机记录在请求乘车事件上送时报“脱机记录重复使用”或“脱机记录id重复”，是什么原因？ 30

# SDK介绍说明

## 术语

|  |  |
| --- | --- |
| 名称 | 说明 |
| 脱机交易 | 脱机交易指POS不在线的交易，本文档特指符合支付宝离线支付规范的脱机交易。 |
| 脱机操作 | 脱机操作特是指用户使用离线二维码在合作伙伴终端进行刷码的行为。一次脱机操作会生成一条脱机操作记录，支付宝通过验证脱机操作的有效性识别脱机交易有效性。根据不同的场景，一笔脱机交易可能包括一笔或者多比脱机操作。例如，单程公交一笔交易包含一个脱机操作，地铁交易通常包括进站和出站两个脱机操作。 |
| 开放平台 | 开放平台是指支付宝的开放平台，支付宝通过开放平台对开发者和合作伙伴输出支付宝业务服务。合作伙伴可通过支付宝开放平台访问支付宝的各种开放业务。  支付宝开放平台可访问<https://openhome.alipay.com/platform/home.htm> |
| POS | Point Of Sale缩写，指合作伙伴收银终端,公交地铁中通常为公交、地铁上下车或者进出站的闸机。 |
| 支付宝离线SDK | 支付宝离线业务终端SDK，支持公交地铁离线进行二维码验证受理。 |
| 卡包 | 支付宝客户端内的卡券/卡包服务。 |
| 支付宝协议 | 支付宝定义的离线二维码、脱机记录的数据格式、结构。 |
| 交通部协议 | 交通部定义的离线二维码、脱机记录的数据格式、结构。 |

## 集成内容

### IoT公交地铁SDK

IoT公交地铁SDK提供支付宝离线二维码有效性验证及日志处理服务。合作伙伴通过调用支付宝离线验证相关能力完成用户离线二维码有效性验证，同时需要请求日志处理相关能力写入或获取SDK产生的日志数据，获取到的日志数据可上送到支付宝开放平台。

支付宝根据软硬件技术规范提供Linux，Android等通用平台的支付宝离线验证SDK。特殊平台SDK需求可联系支付宝技术团队咨询和支持。

合作伙伴通过本地函数调用集成支付宝离线验证SDK，验证支付宝离线二维码。库文件、接口说明请参考[3](#_调用接口说明)。

# SDK安装说明

## 支付宝离线验证SDK

支付宝离线验证SDK支持支付宝离线码技术规范定义的离线验证服务。

支付宝离线验证SDK支持交通部二维码技术规范定义的离线验证服务。

支付宝离线验证SDK支持重庆一码通技术规范定义的离线验证服务。

### 集成说明

#### 环境说明

|  |  |  |
| --- | --- | --- |
| 环境 | 说明 | 备注 |
| ABI | arm/x86/x86\_64 | 目前支付宝提供arm/x86/x86\_64三种体系结构库 |
| OS | linux | 目前支付宝提供Linux系统的SDK |

【注】

特有平台需求请联系支付宝技术团队获取技术支持。

#### 文件说明

首次安装IoT公交地铁SDK时，请先运行install.sh将对应的文件安装到约定目录，另外需要运行service.sh运行后台进程，并集成支付宝离线SDK动态库文件libposoffline.so。集成方法请参考下述头文件。使用方式请参考样例文件example.c。

* libposoffline.so

支付宝离线动态库提供的接口集成在libposoffline.so中，接口的使用示例请参考example.c。

* libalipay\_unisdk.so

日志相关功能提供的接口集成在libalipay\_unisdk.so中，接口的使用示例请参考example.c。

* pos\_crypto.h

本头文件定义了使用支付宝离线动态库需要的数据结构与接口名。详细信息请参考头文件中的注释。

* install.sh

将alipay\_iotd和service.sh拷贝到约定的目录，并且会创建必要的目录。

（约定的目录必须具备读写权限，且重启后不会丢失）

* alipay\_iotd

主要的业务逻辑实现，通过独立进程实现日志数据的存储以及查询等功能。

* service.sh

执行此脚本文件以通过标准方式运行alipay\_iotd。

* alipay\_sdk\_bus.h

提供日志能力的业务逻辑接口描述。

* alipay\_sdk\_client.h

提供iot能力初始化、版本查询的接口描述。

* alipay\_iot\_dict.h

key-value数据结构的C语言实现结构体描述。

* alipay\_iot\_common.h

公共的宏定义。

* example

可以运行的example可执行文件。

* example.c

本文件展示了一个使用IoT公交地铁SDK的简单例程。

#### 集成时序



# 调用接口说明

### 离线验证初始化

#### 业务场景概述

在POS集成使用SDK提供的离线验证功能前，需要进行离线验证初始化，完成必要的数据初始化。例如需要使用的协议标准、支持机构、支持卡类型等，多线程场景下，仅需执行一次。

#### 接口使用概述

仅需调用一次，多次调用报错。

当停止使用时需要调用反初始化接口释放资源，反初始化接口参考3.1.2。

#### 业务执行前提

确定支持的协议标准、机构号、卡类型等。

#### 业务规则

执行任何功能前必须执行初始化。

#### 函数接口

int init(INIT\_REQUEST\* request);

#### 入参

INIT\_REQUEST结构：

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| code\_issuer\_infos | 发码机构信息列表 | INIT\_INFO结构体 | 准备初始化的发码机构信息列表 | 不可空 |  |
| code\_issuer\_info\_number | 发码机构个数 | int | 列表中发码机构个数 | 不可空 | 1 |

INIT\_INFO结构：

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| proto\_type | 协议类型 | char\* | 协议类型 | 不可空 | ALIPAY:支付宝协议  MOT:交通部协议 |
| code\_issuer\_no | 发码机构号 | char\* | hex字符串格式的支付宝或交通部授权的发码机构号 | 不可空 | “00112233” |
| card\_types | 卡类型列表 | char\*\* | 卡类型列表，协议类型为支付宝协议时不可空 | 可空 |  |
| card\_type\_number | 卡类型个数 | int | 卡类型列表中卡类型个数，协议类型为支付宝协议时不可空 | 可空 | 1 |

#### 返回值

返回成功或错误码，定义见2.1.6错误码说明。

#### 样例代码

见example.c。

### SDK反初始化

#### 业务场景概述

不需要使用离线SDK验证功能时调用此接口释放资源，多线程场景下，仅需执行一次。

#### 接口使用概述

仅需调用一次，多次调用报错。

#### 业务执行前提

已调用离线SDK初始化。

#### 业务规则

无

#### 函数接口

int uninit();

#### 入参

无

#### 返回值

成功 或 错误码。

#### 样例代码

见example.c。

### 码信息解析

#### 业务场景概述

识别二维码后，调用本接口解析二维码中的信息。接口只进行二维码格式解析，不做有效性验证（例如过期时间、签名等，验证请使用接口2.1.5）。

若二维码格式非法，则返回相应错误码，若二维码格式合法，则返回二维码对应的业务信息，返回的信息见返回结构体。

支持多线程调用。

#### 接口使用概述

函数调用。

#### 业务执行前提

初始化成功。

#### 业务规则

无。

#### 函数接口

int get\_qrcode\_info(INFO\_REQUEST\* request,

INFO\_RESPONSE\* response);

#### 入参

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| request | 验证请求 | INFO\_REQUEST结构体 | 获取二维码信息的请求 | 不可空 | - |
| response | 验证响应 | INFO\_RESPONSE结构体 | 获取二维码信息的响应 | 不可空 | - |

其中INFO\_REQUEST如下

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| qrcode | 二进制二维码 | unsigned char\* | 用户原始二进制二维码信息 | 不可空 | {0x01,0x02…} |
| qrcode\_len | 二维码长度 | int | 用户原始二进制二维码的长度 | 不可空 | 195 |

其中INFO\_RESPONSE如下

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| proto\_type | 协议类型 | char\* | 协议类型 | 不可空 | “ALIPAY” – 支付宝协议  “MOT” – 交通部协议 |
| code\_info | 码信息 | CODE\_INFO共用体 | 共用体  请根据协议类型选择结构类型 | 不可空 | - |

支付宝码信息

alipay\_code\_info

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| alg\_id | 算法标识 | int | 算法标识 | 不可空 | 1 |
| key\_id | 密钥ID | int | 密钥id | 不可空 | 1 |
| code\_isser\_no | 发码机构编号 | char\* | hex格式发码机构号 | 不可空 | “00000000” |
| code\_issuer\_no\_len | 发码机构号长度 | int | 存放发码机构号的buffer长度 | 不可空 | 9 |
| user\_id | 用户id | char\* | 用户id | 不可空 | 2088xxxxxxxxxxxx |
| user\_id\_len | 用户id长度 | int | 存放用户id的buffer长度 | 不可空 | 17 |
| card\_type | 卡类型 | char\* | 卡类型 | 不可空 | “GJ000001” |
| card\_type\_len | 卡类型长度 | int | 存放卡类型的buffer长度 | 不可空 | 9 |
| card\_no | 卡号 | char\* | 卡号 | 不可空 | - |
| card\_no\_len | 卡号长度 | int | 存放卡号的buffer长度 | 不可空 | 65 |
| card\_data | 卡数据 | unsigned char\* | 二进制格式原始卡数据 | 不可空 | 00010203… |
| card\_data\_len | 卡数据长度 | int | 存放卡数据的buffer长度 | 不可空 | 129 |

交通部码信息

mot\_code\_info

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| sign\_mode | 签名模式 | int | 二维码签名模式  支持  1: 两级SM2签名  2: 三级SM2签名 | 不可空 | 1 |
| cert\_sn | 证书编号 | char\* | 证书编号 | 不可空 | “000001” |
| cert\_sn\_len | 证书编号长度 | int | 证书编号长度 | 不可空 | 7 |
| ca\_key\_idx | 根密钥索引 | int | 根密钥索引  两级SM2模式此字段无效 | 可空 | 1 |
| card\_issuer\_no | 发卡机构编号 | char\* | hex格式发卡机构号 | 不可空 | “000102” |
| card\_issuer\_no\_len | 发卡机构编号长度 | int | 存放发卡机构编号的buffer长度 | 不可空 | 9 |
| code\_isser\_no | 发码机构编号 | char\* | hex格式发码机构号 | 不可空 | “00010203” |
| code\_issuer\_no\_len | 发码机构号长度 | int | 存放发码机构编号的buffer长度 | 不可空 | 9 |
| user\_id | 用户id | char\* | 用户id | 不可空 | 2088xxxxxxxxxxxx |
| user\_id\_len | 用户id长度 | int | 存放用户id的buffer长度 | 不可空 | 17 |
| card\_biz\_type | 卡业务类型 | int | 卡业务类型 | 不可空 |  |
| card\_no | 卡号 | char\* | 卡号 | 不可空 |  |
| card\_no\_len | 卡号长度 | int | 存放卡号的buffer长度 | 不可空 | 65 |
| card\_data | 卡数据 | unsigned char\* | 二进制格式原始卡数据 | 不可空 | “00010203…” |
| card\_data\_len | 卡数据长度 | int | 存放卡数据的buffer长度 | 不可空 | 129 |

#### 返回值

返回成功或错误码，错误码见2.1.6错误码说明。

#### 样例代码

见example.c。

### 码信息验证

#### 业务场景概述

本接口验证二维码是否有效验，对于验证合法的二维码，SDK自动生成脱机操作记录，SDK会对脱机操作记录执行签名。脱机交易结算时，支付宝将验证原始码信息合法性以及签名有效性。

二维码验证通过后，记录脱机操作记录，完成本地记账并提示刷码通过。

调用本接口需要提供验证密钥，密钥可通过2.1.4码信息解析的信息提取匹配的密钥。

#### 接口使用概述

函数调用。

#### 业务执行前提

识别用户二维码，并已经执行过支付宝离线验证SDK初始化。

#### 业务规则

无。

#### 函数接口

int verify\_qrcode\_v3(VERIFY\_REQUEST\_V3\* request\_v3, VERIFY\_RESPONSE\_V3\* response\_v3);

#### 入参

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| request\_v3 | 验证请求 | VERIFY\_REQUEST\_V3结构体 | 验证用户二维码的请求 | 不可空 | - |
| response\_v3 | 验证响应 | VERIFY\_RESPONSE\_V3结构体 | 验证用户二维码的响应 | 不可空 | - |

VERIFY\_REQUEST\_V3结构说明

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| qrcode | 二进制二维码 | unsigned char\* | 用户原始二进制二维码信息 | 不可空 | {0x01,0x02…} |
| qrcode\_len | 二维码长度 | int | 用户原始二进制二维码的长度 | 不可空 | 195 |
| key | 密钥 | char\* | hexString格式的密钥信息 | 不可空 | “03E0CEE489BDCA42CAA4C09E42CCBA341CE2BF20C4338958D7939F7087744E2620” |
| pos\_param | 合作伙伴pos信息 | POS\_PARAM\_STRUCT结构体 | 合作伙伴pos相关参数信息，见POS\_PARAM\_STRUCT结构说明 | 不可空 | - |

合作伙伴pos信息

POS\_PARAM\_STRUCT

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 | |
| --- | --- | --- | --- | --- | --- | --- |
| record\_id | 脱机记录流水号 | char\* | 脱机记录流水号，唯一 | 不可空 | - | |
| consumption\_type | 消费类型 | int | 消费类型 | 不可空 | 0 – 单次消费  1 – 复合消费 | |
| pos\_id | 终端号 | char\* | 机构下唯一的终端号 | 不可空 | 20170000000001 | |
| pos\_mf\_id | 终端厂商编号 | char\* | 由检查机构分配的终端厂商编号 | 不可空 | 9998112123 | |
| pos\_sw\_version | 终端软件编号 | char\* | 终端软件版本号 | 不可空 | 2.6.14.03arm | |
| merchant\_type | 合作伙伴类型 | char\* | 合作伙伴类型 | 不可空 |  | |
| currency | 币种 | char\* | 币种，人民币为156 | 可空 | 156 | |
| amount | 金额 | char\* | 消费金额，单位分 | 可空 | 消费类型为0时必传 | |
| bus\_no | 车辆号 | char\* | 车辆编号 | 可空 | 123123 | |
| plate\_no | 车牌号 | char\* | 车牌号 | 可空 | 京A00001 | |
| driver\_no | 司机号 | char\* | 司机号 | 可空 |  | |
| line\_info | 线路信息 | char\* | 线路信息 | 可空 |  | |
| station\_no | 站点编号 | char\* | 站点编号 | 可空 |  | |
| lbs\_info | 经纬度信息 | char\* | 经纬度信息 | 可空 |  |
| record\_type | 脱机记录类型 | char\* | 脱机记录类型 | 可空 | 支持支付宝协议时，请传入对应类型。  BUS – 公交  SUBWAY – 地铁 |

VERIFY\_RESPONSE结构说明

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| record | 脱机记录 | char\* | 存放自动生成的脱机操作记录，记录数据为hex格式字符串，  合作伙伴需要自行为该数组分配内存、回收内存。 | 不可空 | - |
| record\_len | 脱机记录空间大小 | int | 存放record字符串的空间大小。  验证完毕后，会被修改为真实record的长度。 | 不可空 | 2048 |

#### 返回值

成功或错误码，错误码参见2.1.6。

#### 样例代码

见example.c。

### 错误码说明

|  |  |  |  |
| --- | --- | --- | --- |
| 错误码 | 值 | 错误描述 | 解决方案 |
| MALFORMED\_QRCODE | -1 | 二维码格式错误 | 二维码格式错误，检查输入的二维码是否正确。 |
| QRCODE\_INFO\_EXPIRED | -2 | 二维码信息失效 | 用户离线公交码失效，提示用户刷新二维码。 |
| QRCODE\_KEY\_EXPIRED | -3 | 二维码密钥失效 | 用户离线公交码授权秘钥失效，提示用户联网刷新离线公交码后再使用。 |
| POS\_PARAM\_ERROR | -4 | pos\_param错误 | 输入的POS信息参数错误，检查输入的pos\_param是否正确。 |
| QUOTA\_EXCEEDED | -5 | 用户单笔额度超限 | 用户单笔限额超限，提示用户由于额度超限。 |
| NO\_ENOUGH\_MEMORY | -6 | 内存不足 | 内存申请失败，程序运行内存不足。 |
| SYSTEM\_ERROR | -7 | 系统内部错误 | 内部处理发生异常，通常不会发生。遇到内部错误，可联系支付宝技术排查。 |
| CARDTYPE\_UNSUPPORTED | -8 | 卡类型不支持 | POS不支持二维码对应的卡类型 |
| NOT\_INITIALIZED | -9 | 未执行初始化 | 未执行初始化，请先执行初始化 |
| ILLEGAL\_PARAM | -10 | 参数错误 | 入参错误导致的失败 |
| PROTO\_UNSUPPORTED | -11 | 不支持的协议版本 | 当前使用sdk不支持二维码的协议版本 |
| QRCODE\_DUPLICATED | -12 | 重复的二维码 | 使用已经验证成功的二维码重复刷码 |
| INSTITUTION\_NOT\_SUPPORT | -13 | 机构不支持 | 不支持二维码中指定的机构 |
| INIT\_DUPLICATED | -14 | 重复初始化 | 已经执行过初始化，不允许多次执行 |

### IoT日志功能初始化

#### 业务场景概述

在POS开启SDK提供的日志采集功能前，需要进行IoT设备初始化，完成必要的数据初始化。需要传入的参数包括需要合作伙伴唯一标示和设备唯一序列号。

#### 接口使用概述

仅需调用一次，多次调用无效。

#### 业务执行前提

请联系支付宝相关人员确认合作伙伴唯一标示，请保证设备唯一序列号在合作伙伴标示下唯一。

#### 业务规则

执行任何日志相关功能前必须执行日志功能初始化。

#### 函数接口

void alipay\_iot\_init(const char\* isvid, const char\* sn);

#### 入参

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| isvid | 合作伙伴唯一标示 | const char\* | 支付宝分配的合作伙伴唯一标示 | 不可空 | 1234567890 |
| sn | 设备唯一序列号 | const char\* | 合作伙伴标示下的设备唯一序列号 | 不可空 | “00112233” |

#### 返回值

无

#### 样例代码

见example.c。

### IoT设备控制

#### 业务场景概述

* 在机具开启SDK提供的日志采集功能后，可以通过IoT设备控制接口控制日志采集行为是否开启。
* 在收到需下发给机具的数据后，请通过此接口将下发数据发送至SDK中。

#### 接口使用概述

希望进行SDK行为控制时，调用此接口。

#### 业务执行前提

请先执行日志功能初始化。

#### 业务规则

执行任何日志相关功能前必须执行日志功能初始化。若收到下发数据，只需使用下发数据调用一次此接口。

#### 函数接口

int alipay\_iot\_control(int type, ...);

#### 入参

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| type | 操作类型 | int | 请求机具进行的操作类型 | 不可空 | ALIPAY\_IOT\_GET\_STATUS  ALIPAY\_IOT\_SET\_STATUS  等 |
| ... | 操作内容 | 可变参数 | 实际请求操作的内容 | 不可空 | “00112233” |

#### 返回值

返回成功或错误码，定义见日志采集模块错误码说明。

#### 样例代码

见example.c。

#### DEMO示例

设置日志采集/写入开关为开启

​int rv = alipay\_iot\_control(ALIPAY\_IOT\_SET\_STATUS,1);

获取日志开关状态

int status = 0;

int rv = alipay\_iot\_control(ALIPAY\_IOT\_GET\_STATUS, &status);

### 业务信息配置

#### 业务场景概述

机具执行初始化后，必须进行业务信息配置。需要配置的内容请参考接口描述。

业务信息配置后，所有写入的日志会自动将业务信息注入日志数据，更有利于数据的分析。

如要求日志中需要添加车牌号、城市、线路号、票价等业务信息，可使用此接口注册，日志生成时会自动绑定业务信息。

#### 接口使用概述

生命周期内可多次调用，同配置多次设置将覆盖原有配置。

#### 业务执行前提

* 日志能力初始化后，请调用此接口。
* 机具业务配置发生变更，请调用此接口。

#### 业务规则

执行任何日志相关功能前必须执行日志功能初始化。

#### 函数接口

int alipay\_iot\_set\_config(const aliot\_dict\* configs);

#### 入参

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| configs | 执行配置的内容 | const aliot\_dict\* | 执行配置的内容，包括票价、实扣、设备id、车牌号、支持线路、公司信息等。 | 不可空 | {“price”: “2.00”...} |

字典内容如下：

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| ticket\_price | 票价 | const char\* | 乘坐此交通工具的票面价格(不包括折扣) | 不可空 | “ticket\_price”: “2.00” |
| actual\_price | 实扣 | const char\* | 机具扫码后实际要扣用户账户的金额 | 不可空 | “actual\_price”: “2.00” |
| vehicle\_no | 车号 | const char\* | 车牌号 | 不可空 | “vehicle\_no”:”0023” |
| line | 线路号 | const char\* | 线路号 | 不可空 | “line”:”192路BRT” |
| company | 公司名称 | const char\* | 公交运营公司 | 不可空 | “company”:”北京市交通公司” |
| city | 城市 | const char\* | 城市信息 | 不可空 | “city”:”北京” |
| driver\_id | 司机号 | const char\* | 司机号 | 不可空 | “driver\_id”:”1234576” |
| device\_id | 设备编号 | const char\* | 设备的唯一SN | 不可空 | “device\_id”:”1234576” |
| supplier\_id | 厂商编号 | const char\* | 厂商编号（支付宝申请分配） | 不可空 | supplier\_id:”1234567” |
| item\_id | 设备类型ID | const char\* | 设备类型ID（支付宝申请分配） | 不可空 | item\_id:”11235678” |
| area | 城区 | const char\* | 区域信息 | 可空 | “area”:”东城区” |
| plate\_no | 车牌号 | const char\* | 车牌号 | 可空 | “plate\_no”:”京A 123456” |

注：deviceid就是posid，厂商必须保证公交计价事件流程中上报的posid和sdk链路上报的deviceid相同

#### 返回值

返回成功或错误码，定义见日志采集模块错误码说明。

#### 样例代码

见example.c。

#### DEMO示例

写入车牌和城市业务配置信息

aliot\_dict\* dict = aliot\_dict\_new();

aliot\_dict\_set(dict, "device\_id", "AAB38069");

aliot\_dict\_set(dict, " supplier\_id", "111112332");

aliot\_dict\_set(dict, "item\_id", "ghsdsds");

aliot\_dict\_set(dict, " ticket\_price", "2.00");

int rv = alipay\_iot\_set\_config(dict);

aliot\_dict\_del(dict);

### 日志同步[可选]

#### 业务场景概述

构造符合对应格式的数据请求此接口，会将日志数据落地到SDK中，在后续发起日志采集请求时，会采集到此日志。

#### 接口使用概述

希望将日志数据落地到SDK中待后续日志采集捞取时，请调用此接口。

#### 业务执行前提

请先执行日志功能初始化且通过IoT设备控制打开日志采集功能。

#### 业务规则

请按照接口文件描述的格式请求此接口。

#### 函数接口

int alipay\_iot\_data\_store(const extra\_params\* params, const aliot\_dict\* datas);

#### 入参

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| params | 扩展参数 | extra\_params\* | 插入数据的额外配置参数，包括产品id、业务类型与子业务类型三个，方便查询使用，不使用可直接为空。 | 可空 | NULL |
| datas | 希望被存储的KV形式数据 | aliot\_dict\* | 实际请求存储到日志数据库的内容 | 不可空 | “card\_type=ANT00001...” |

#### 返回值

返回成功或错误码，定义见日志采集模块错误码说明。

#### 样例代码

见example.c。

### 日志采集

#### 业务场景概述

请先进行机具执行初始化，以及必要的业务信息配置。打开日志采集开关后，请求此接口采集已经产生的日志。

该接口可获取设备交易异常/系统异常等信息，使用该接口定时上送日志，便于设备异常问题快速定位解决。

#### 接口使用概述

可能需要多次调用，根据响应中返回的参数判断是否需要继续获取日志。需要获取日志的情况下，请持续执行以获取全部日志。

获取的日志已经过ZIP压缩，可直接透传给云服务器。

如果需要查看压缩的数据，可调用show\_unzip\_data(example.c)。

#### 业务执行前提

完成日志能力初始化且打开日志采集功能。

#### 业务规则

执行任何日志相关功能前必须执行日志功能初始化，请设定一定的时间间隔请求此接口以获取最新的日志，每次请求请循环请求直到日志全部上送完毕。

【注】该接口调用后，必须调用alipay\_iot\_data\_free释放out\_data空间。alipay\_iot\_data\_free的调用说明见[3.1.10](#_日志采集内存释放)

#### 函数接口

int alipay\_iot\_data\_query(const query\_params\* params, char\*\* out\_data, int\* out\_len);

#### 入参

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| params | 执行查询的配置 | const query\_params\* | 执行查询的配置，包括用来增加查询条件的extra\_params，描述最大输出大小的max\_data\_size(最大为4096) | 不可空 | query\_params {  params=NULL;  max\_data\_size=4096;} |
| out\_data | 输出的数据 | char\*\* | 失败时返回NULL | 不可空 | - |
| out\_len | 输出数据的长度 | int\* | 返回的结果长度 | 不可空 | 10 |

#### 返回值

返回成功或错误码，定义见日志采集模块错误码说明。

#### 样例代码

见example.c。

#### DEMO示例

void TimerGetLogFunc()

{

char\* out\_data = NULL;

int out\_len = 0;

alipay\_iot\_query\_params params;

params.max\_data\_size = 4096;

params.extras.product\_id = NULL;

params.extras.biz\_type = NULL;

params.extras.sub\_type = NULL;

int rv = alipay\_iot\_data\_query(&params, &out\_data, &out\_len);

if (rv >= 0 && out\_len > 0)

{

//send out\_data to server

}

alipay\_iot\_data\_free(out\_data);

}

### 日志采集内存释放

#### 业务场景概述

执行日志采集后，out\_data会被分配空间，out\_data使用完毕后请调用此接口进行释放。

#### 接口使用概述

调用日志采集后，释放内存时调用。

#### 业务执行前提

成功从日志采集请求中获取到数据并执行完业务逻辑后。

#### 业务规则

请按照接口文件描述的格式请求此接口。

#### 函数接口

void alipay\_iot\_data\_free(void\* out\_data);

#### 入参

| 参数 | 参数名称 | 类型 | 参数说明 | 是否可为空 | 样例 |
| --- | --- | --- | --- | --- | --- |
| out\_data | 日志输出数据 | void\* | 日志输出数据 | 不可空 | - |

#### 返回值

无返回

#### 样例代码

见example.c。

# 常见问题说明

## 执行install.sh报错

在执行install.sh安装sdk到约定目录，会建立sdk运行的必要目录等，需要该目录下的读写权限；请先检查指定的安装目录是否具有读写权限；若需要更改安装目录位置，可联系技术支持重新输出相应的sdk。

## 日志写入后不获取会不会造成空间无限增长？

单条日志不超过1024字节；

最大上限设置为1000条；

最大占用空间为1M；

超过上限条数后，会根据时间排序优先删除时间较早的日志100条；

## 日志查询接口多久调用一次合适？

合作伙伴可根据需要设置定时获取的时间间隔，一般推荐为5Min。

## 日志查询接口获取数据后，需要解析再上传服务器？

日志查询得到的数据为压缩(zip)后的数据，获取数据后直接上送给合作伙伴的云服务器即可，由云服务器和支付宝服务器进行解析和接口数据重封装。

## 是否每次验码之前初始化都需要执行初始化？

机具开机时，初始化一次即可，反初始化在程序正常注销时调用即可。初始化与反初始化之间，可重复请求“获取二维码信息”与“验证二维码”。

## 请求获取二维码信息成功，但请求验证时出现段错误，是什么原因？

公交地铁SDK中集成了openssl安全算法库，如机具原生系统中也存在openssl，可能产生符号冲突，此时需评估本地openssl是否能够删除。如不能删除请联系支付宝工作人员解决此问题。

## 生成的脱机记录在请求乘车事件上送时报“脱机记录重复使用”或“脱机记录id重复”，是什么原因？

同一笔脱机记录只能关联唯一一次乘车事件，请确认当前脱机记录是否已经关联过其他乘车事件。机具在验证二维码时，需要传入参数” record\_id”，请保证此record\_id在合作伙伴id下唯一，否则也将认定为按脱机记录重复。