HONOR Connect IoT SDK接口文档

目录

一、简介

HONOR Connect IoT SDK用于IoT设备开发者将IoT设备注册到荣耀账号下，实现设备的互联互通能力。

二、功能列表

功能	描述
设备发现	提供SoftAP和CoAP发现能力
设备配网	提供设备接收WiFi信息并接入到网络的能力
设备认证	提供固定密钥、软license认证能力
设备注册	提供设备注册到荣耀设备云的能力
设备控制	提供荣耀智慧空间控制IoT设备的能力
OTA升级	提供设备在线升级的能力
恢复出厂	提供HONOR Connect IoT SDK恢复出厂的能力
日志收集	提供HONOR Connect IoT SDK日志和开发者日志收集上传的能力

三、接口列表

3.1 HONOR Connect IoT SDK 初始化去初始化

3.1.1 初始化

接口原型

```
struct MagicLinkDataVal {
    unsigned int dataIdx;
    unsigned int dataLen;
    unsigned char *data;
};

/* 设备控制回调 */
struct MagicLinkCtrlFunc {
    /* 查询设备属性，同步接口
     * 参数out,在接口内部使用malloc分配内存空间，outLen表示分配出的内存空间大小，
     * HONOR Connect IoT SDK调用接口后会自行调用free释放空间
     */
    int (*getServiceProperty)(const char *service, const char *property, void
**out, unsigned int *outLen);

    /* 设置设备属性，返回-100表示正在执行中，执行完成后，通过调用report上报状态信息 */
    int (*setServiceProperty)(const char *service, const char *property, const void
*in, unsigned int inLen);

    /* 执行动作，MagicLink等待动作执行结果 */
    int (*execAction)(const char *service, const char *action, unsigned int
dataNum,
        const struct MagicLinkDataVal *dataArray);
};

/* 初始化 */
int MagicLinkInit(const char *devModelJson, const struct MagicLinkCtrlFunc
*ctrlFunc);
```

入参说明

参数	类型	必选/可选	说明
devModelJson	const char *	必选	设备模型，json格式
ctrlFunc	const struct MagicLinkCtrlFunc *	必选	设备控制回调

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

调用此接口初始化HONOR Connect IoT SDK。

3.1.2 去初始化

接口原型

```
int MagicLinkDeInit(void);
```

入参说明

无

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

调用此接口，HONOR Connect IoT SDK会释放系统资源。

3.2 HONOR Connect IoT SDK状态通知

接口原型

```
/* HONOR Connect IoT SDK状态 */
```

```
enum MagicLinkSDKStatus {
    /* 配网相关 */
    STATUS_NETCFG_ENTER = 100,
    STATUS_NETCFG_RECV_INFO,
    STATUS_NETCFG_TIMEOUT,
    STATUS_NETCFG_STOP,

    /* 连接WiFi相关 */
    STATUS_NET_CONNECTING = 200,
    STATUS_NET_CONNECT_SUCC,
    STATUS_NET_CONNECT_FAIL,

    /* 设备注册相关 */
    STATUS_REGISTER_RECV_INFO = 300,
    STATUS_REGISTING,
    STATUS_REGISTER_SUCC,
    STATUS_REGISTER_FAIL,

    /* 设备注册相关，登录信息过期，刷新登录信息 */
    STATUS_SECRET_TIMEOUT = 400,
    STATUS_SECRET_REFRESHING,
    STATUS_SECRET_REFRESH_FAIL,
    STATUS_SECRET_REFRESH_SUCC,

    /* 登录设备云相关 */
    STATUS_LOGIN_ONLINE = 500,
    STATUS_LOGIN_OFFLINE,
    STATUS_LOGOUT,

    /* 设备删除相关 */
    STATUS_DEVICE_NOT_IN_CLOUD = 600,
    STATUS_DEVICE_DELETE,
    STATUS_DEVICE_RESET,

    /* 设备升级相关 */
    STATUS_OTA_CHKING_VERSION = 700,
    STATUS_OTA_CHK_VERSION_DONE,
    STATUS_OTA_UPGRADING,
    STATUS_OTA_DOWNLOAD_COMPLETE,
    STATUS_OTA_DOWNLOAD_TIMEOUT,
    STATUS_OTA_UPGRADE_SUCC,
    STATUS_OTA_UPGRADE_FAIL,
    STATUS_OTA_MCU_INSTALL_TIMEOUT,

    /* 设备异常相关 */
    STATUS_SYSTEM_ABNORMAL = 1000, /* 系统异常，SDK恢复出厂 */

    STATUS_BUTT
};

/* 查询HONOR Connect IoT SDK状态 */
enum MagicLinkSDKStatus MagicLinkGetStatus(void);

/* 注册HONOR Connect IoT SDK状态回调 */
int MagicLinkRegRecvStatus(void (*recvStatus)(enum MagicLinkSDKStatus status));
```

入参说明

参数	类型	必选/可选	说明
recvStatus	void (*)(enum MagicLinkSDKStatus)	必选	接收设备状态回调。取值列表 0，初始化成功；

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

开发者调用查询接口主动查询；开发者注册回调被动接收。

3.3 设备配网

3.3.1 启动配网

接口原型

```
/* 启动配网 */
int MagicLinkStartNetCfg(void);
```

入参说明

无

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

1）在初始化之后调用；
2）设备未注册时，调用此接口，可实现设备配网，并将设备注册到设备云；
3）设备已注册时，调用此接口，仅实现重新配网功能(重新配网功能需要设备有物理触发机制，例如:按键、组合按键)；

4）配网采用SoftAP模式，超时时间默认为10min，超过10min系统自动退出SoftAP模式，配网结束。可通过设置设备属性接口调整配网超时时间。

3.3.2 停止配网

接口原型

```
/* 终止配网 */
int MagicLinkStopNetCfg(void);
```

入参说明

无

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

1）设备处于配网状态，调用此接口停止配网。

3.3.3 是否已注册

接口原型

```
/* 设备是否已经注册到荣耀设备云 */
bool MagicLinkIsRegistered(void);
```

入参说明

无

出参说明

无

返回说明

类型	取值	说明
bool	true	已注册
	false	未注册

使用说明

1）针对启用配网功能的产品，此接口可作为启动配网的条件。如果已注册，不启动配网；如果未注册，启动配网。

##

3.4 设备控制

3.4.1 查询设备属性

接口原型

```
int (*getServiceProperty)(const char *service, const char *property, void **out,
unsigned int *outLen);
```

入参说明

参数	类型	必选/可选	说明
service	const char *	必选	查询的服务
property	const char *	必选	查询的属性

出参说明

参数	类型	必选/可选	说明
out	void **	必选	存储结果空间，集成方调用malloc接口分配
outLen	unsigned int *	必选	分配的out的的大小

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

开发者在初始化阶段注册，满足设备模型定义。

3.4.2 设置设备属性

接口原型

```
int (*setServiceProperty)(const char *service, const char *property, const void
*in, unsigned int inLen);
```

入参说明

参数	类型	必选/可选	说明
service	const char *	必选	设置的服务
property	const char *	必选	设置的属性
in	const void *	必选	设置的值
inLen	unsigned int	必选	值占用的字节数

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	-100	正在执行，HONOR Connect IoT SDK不会立刻上报设备最新状态，集成方需要在执行完成后调用report接口上报最新状态。
	其他	失败

使用说明

开发者在初始化阶段注册，满足设备模型定义。

3.4.3 设备执行动作

接口原型

```
int (*execAction)(const char *service, const char *action, unsigned int dataNum,
const struct MagicLinkDataVal *dataArray);
```

入参说明

参数	类型	必选/可选	说明
service	const char *	必选	执行动作的服务
action	const char *	必选	执行的动作
dataNum	unsigned int	必选	*dataArray 数组个数
dataArray	const struct MagicLinkDataVal[]	必选	action执行数据

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

开发者在初始化阶段注册，满足设备模型定义。

3.4.4 设备状态上报

接口原型

```
int MagicLinkReportServiceStatus(const char *service);
```

入参说明

参数	类型	必选/可选	说明
service	const char *	必选	上报的服务，一次上报所有属性值

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

设备状态发生变化，开发者调用此接口上报服务下的所有属性最新状态。

接口原型

```
int MagicLinkReportPropertyStatus(const char *service, unsigned int propNum, const unsigned int *propIndexArray);
```

入参说明

参数	类型	必选/可选	说明
service	const char *	必选	上报的服务
propNum	unsigned int	必选	上报的属性数量
propIndexArray	const unsigned int *	必选	上报的属性的index数组

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

设备状态发生变化，开发者调用此接口上报设备服务下的某个或者某几个属性的最新状态。

3.4.5 设备事件上报

接口原型

```
int MagicLinkReportServiceEvent(const char *service, const char *eventId, unsigned
int dataNum, const struct MagicLinkDataVal *dataArray);
```

入参说明

参数	类型	必选/可选	说明
service	const char *	必选	上报的服务
eventId	const char *	必选	事件ID
dataNum	unsigned int	必选	上报的事件的value数量
dataArray	struct MagicLinkDataVal []	必选	事件数据，根据设备模型定义填写。

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

设备发送事件，开发者调用此接口上报设备事件。

3.5 恢复出厂

3.5.1 恢复出厂

接口原型

```
int MagicLinkReset(void);
```

入参说明

无

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	-102	正在恢复出厂中
	其他	失败

使用说明

1）调用此接口执行HONOR Connect IoT SDK恢复出厂；
2）调用此接口，HONOR Connect IoT SDK会释放系统资源，清理磁盘保存的数据，并重新进入初始化状态；
3）返回-102时，HONOR Connect IoT SDK正在执行恢复出厂操作，开发者无需重复执行。

补充：此接口只完成*HONOR Connect IoT SDK*相关资源的释放，在恢复出厂时开发者需要完成除*HONOR Connect IoT SDK*以外其他资源的释放。

3.6 OTA

3.6.1 模组固件升级（仅linux系统适用）

接口原型

```
typedef enum {
    OTA_PACKAGE_BUF_FIRST,
    OTA_PACKAGE_BUF_MIDDLE,
    OTA_PACKAGE_BUF_LAST,
    OTA_PACKAGE_BUF_BUTT
} OtaPackageBufType;
```

```
typedef struct MagicLinkOtaFunc {
    /* 存储版本包回调 */
    int (*writeOtaPkg)(const void *position, unsigned int offset, const void *data,
unsigned int len, OtaPackageBufType type);

    /* 执行升级动作 */
    int (*execOta)(void);

    /* 获取版本包储存空间大小 */
    int (*getOtaPkgSaveSize)(unsigned int *size);
} MagicLinkOtaFunc;

/* 注册OTA升级回调 */
int MagicLinkRegOtaFunc(const MagicLinkOtaFunc *otaFunc);
```

入参说明

参数	类型	必选/可选	说明
ctrlFunc	const struct MagicLinkCtrlFunc *	必选	ota回调

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

开发者实现MagicLinkOtaFunc结构体的函数，通过该接口注册到SDK。

3.6.1.1 存储版本包

接口原型

```
int (*writeOtaPkg)(const void *position, unsigned int offset, const void *data,
unsigned int len, OtaPackageBufType type);
```

入参说明

参数	类型	必选/可选	说明
position	const void *	必选	版本包存储位置
offset	unsigned int	必选	版本包存储偏移
data	void *	必选	版本包数据
len	unsigned int	必选	版本包数据长度
type	OtaPackageBufType	必选	版本包数据类型

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

HONOR Connect IoT SDK调用此回调存储OTA版本包。
1）存储版本包首包数据时OtaPackageBufType使用OTA_PACKAGE_BUF_FIRST；
2）存储版本包非首包数据时OtaPackageBufType使用OTA_PACKAGE_BUF_MIDDLE；
3）版本包存储完成以后，HONOR Connect IoT SDK使用OTA_PACKAGE_BUF_LAST存储一次空数据，表示版本包存储完成；
4）版本包下载失败时，OtaPackageBufType使用OTA_PACKAGE_BUF_BUTT。

3.6.1.2 执行升级动作

接口原型

```
int (*execOta)(void);
```

入参说明

无

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

版本包下载完成后，HONOR Connect IoT SDK调用此回调，完成版本包安装。开发者需要在安装动作执行完成以后重启系统或者重启执行SDK的进程。在重启后重新连接到设备云时，SDK会校验升级以后的版本号是否与升级目标版本号一致，一致则判断升级成功，不一致则判断升级失败。

3.6.1.3 获取版本包储存空间大小

接口原型

```
int (*getOtaPkgSaveSize)(unsigned int *size);
```

入参说明

无

出参说明

参数	类型	必选/可选	说明
size	unsigned int *	必选	版本包存储空间大小

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

HONOR Connect IoT SDK调用此回调获取版本包储存空间大小。

3.6.2 设备MCU升级

接口原型

```
typedef struct MagicLinkMcuOtaFunc {
    /* 开始MCU升级 */
    int (*mcuOtaStart)(const char *version, const char *pkgName, unsigned int
pkgSize, const unsigned char *sha256,
        unsigned int sha256Len);
    /* 发送MCU版本包 */
    int (*mcuOtaSendPkg)(unsigned int offset, const unsigned char *pkgData,
unsigned int pkgDataLen);
    /* 通知版本包发送完成 */
    int (*mcuOtaEnd)(void);
    /* 通知MCU OTA过程异常 */
    void (*notifyMcuOtaException)(int status, const char *data);
} MagicLinkMcuOtaFunc;

/* 注册MCU升级回调 */
int MagicLinkRegMcuOtaFunc(const MagicLinkMcuOtaFunc *mcuRegFunc);
```

入参说明

参数	类型	必选/可选	说明
mcuRegFunc	const struct MagicLinkMcuOtaFunc*	必选	MCU回调

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

开发者实现MagicLinkMcuOtaFunc结构体的函数，通过该接口注册到HONOR Connect IoT SDK。

3.6.2.1 开始MCU升级

接口原型

```
int (*mcuOtaStart)(const char *version, const char *pkgName, unsigned int pkgSize,
const unsigned char *sha256,
    unsigned int sha256Len);
```

入参说明

参数	类型	必选/可选	说明
version	const char *	必选	新MCU版本号
pkgName	const char *	必选	新MCU版本包名称
pkgSize	unsigned int	必选	新MCU版本包长度
sha256	const unsigned char *	必选	新MCU版本包sha256值
sha256Len	unsigned int	必选	新MCU版本包sha256值长度

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

HONOR Connect IoT SDK通知MCU开始升级。

3.6.2.2 发送MCU版本包

接口原型

```
int (*mcuOtaSendPkg)(unsigned int offset, const unsigned char *pkgData, unsigned
int pkgDataLen);
```

入参说明

参数	类型	必选/可选	说明
offset	unsigned int	必选	MCU版本包偏移
pkgData	const unsigned char *	必选	MCU版本包数据
pkgDataLen	unsigned int	必选	MCU版本包数据长度

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

HONOR Connect IoT SDK向MCU传递版本包数据。

3.6.2.3 通知MCU版本包发送完成

接口原型

```
int (*mcuOtaEnd)(void);
```

入参说明

无

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

MCU版本包数据发送完成后，HONOR Connect IoT SDK调用此回调通知MCU侧版本包发送完成，可以启动MCU升级安装。

3.6.2.4 通知MCU升级过程异常

接口原型

```
void (*notifyMcuOtaException)(int status, const char *str);
```

入参说明

参数	类型	必选/可选	说明
status	int	必选	状态码
str	const char *	必选	异常说明

出参说明

无

返回说明

```
无
```

使用说明

HONOR Connect IoT SDK通知MCU升级过程异常，状态码和异常说明如下：

状态码	异常说明
-1	download failed
-1	MCU upgrade failed

3.6.3 MCU通知HONOR Connect IoT SDK升级状态

接口原型

```
/* MCU通知SDK升级状态 */
int MagicLinkSetMcuOtaStatus(int status, const char *data);
```

入参说明

参数	类型	必选/可选	说明
status	int	必选	状态码
data	const char *	必选	异常说明

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

MCU升级成功或者升级发生异常时调用该接口通知HONOR Connect IoT SDK。状态码和异常说明如下：

状态码	异常说明
0	MCU upgrade success
-1	MCU upgrade failed

3.7 Wifi接口（仅linux系统适用）

3.7.1 注册网络接口

接口原型

```
/* STA配置类型 */
typedef struct {
    unsigned char ssid[MAGICLINK_MAX_LEN_SSID]; /* 长度为33，包含字符串结束符 */
    unsigned char password[MAGICLINK_MAX_LEN_PASSWORD];  /* 长度65，包含字符串结束符
*/
} MagicLinkStaConfig;

/* AP配置类型 */
typedef struct {
    unsigned char ssid[MAGICLINK_MAX_LEN_SSID]; /* 长度为33，包含字符串结束符 */
    unsigned char password[MAGICLINK_MAX_LEN_PASSWORD]; /* 长度为65，包含字符串结束符
*/
    unsigned char channel;
    unsigned char authMode;
    unsigned char ssidHidden;
    unsigned char maxConnection;
} MagicLinkSoftapConfig;

typedef struct {
```

```
    int (*wifiInit)(void);
    int (*wifiDeinit)(void);
    /* 须确保在初始状态和station模式能够打开ap热点 */
    int (*createSoftap)(const MagicLinkSoftapConfig *config);
    int (*closeSoftap)(void);
    /* ssid和pwd任意一个错误，或者链接异常时通过返回值反馈，0代表正常 */
    int (*wifiConnect)(const MagicLinkStaConfig *config);
    int (*wifiConnectInfoClean)(void);
    int (*wifiDisconnect)(void);
} WiFiAdapterFunc;

/* 注册网络回调函数 */
int MagicLinkRegWifiFunc(const WiFiAdapterFunc *func);
```

入参说明

参数	类型	必选/可选	说明
func	const WiFiAdapterFunc *	必选	Wifi适配的函数指针列表

出参说明

```
无
```

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

实现WiFiAdapterFunc函数指针对应的函数接口，通过该接口注册。

3.7.2 WiFi初始化

接口原型

```
/* wifi初始化 */
int (*wifiInit)();
```

入参说明

无

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

无

使用说明

调用此接口初始化wifi模块

3.7.3 WiFi去初始化

接口原型

```
/* wifi去初始化 */
int (*wifiDeinit)();
```

入参说明

参数	类型	必选/可选	说明
wlanName	const char *	必选	网卡名称

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

调用此接口去初始化设备wifi模块，清除已保存wifi信息

3.7.4 创建SoftAP

接口原型

```
/* 创建SoftAP */
int (*createSoftap)(const MagicLinkSoftapConfig *config);
```

入参说明

参数	类型	必选/可选	说明
config	const MagicLinkSoftApConfig *	必选	AP配置数据

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

无

使用说明

要求在SoftAP和Sta模式都可调用该接口创建SoftAP。

3.7.5 关闭SoftAP

接口原型

```
/* 关闭SoftAP */
int (*closeSoftap)();
```

入参说明

无

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

无

使用说明

在SoftAP模式下调用该接口关闭Wifi模组的SoftAP

3.7.6 连接WiFi

接口原型

```
/* WiFi连接 */
int (*wifiConnect)(const MagicLinkStaConfig *config);
```

入参说明

参数	类型	必选/可选	说明
config	const MagicLinkStaConfig *	必选	STA配置数据

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

无

使用说明

1）调用此接口设备连接wifi。要求在连接wifi失败时的各种异常场景返回失败。例如：ssid错误，password错误等异常场景。

2）连接wifi失败之后，HONOR Connect IoT SDK会通过状态通知接口MagicLinkRegRecvStatus将STATUS_NET_CONNECT_FAIL的状态通知给开发者，

开发者可以根据该状态决定如何提示用户（例如：采用灯光颜色、闪灯模式、语音等）并决定采用何种模式（例如：按键、重新上电、超时自动）调用MagicLinkStartNetCfg接口重新进入发现配网模式。

3）连接wifi成功之后，开发者需要将wifi的ssid和password设置到芯片中，并使能芯片在STA模式自动重连wifi功能。

3.7.7 清除WiFi连接信息

接口原型

```
/* WiFi连接 */
int (*wifiConnectInfoClean)(void);
```

入参说明

```
无
```

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

调用此接口清除已保存wifi的ssid和password。

3.7.8 断连WiFi

接口原型

```
/* 断开连接 */
int (*wifiDisconnect)(void);
```

入参说明

无

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

调用此接口设备断连wifi。

3.8 P2P接口（仅linux系统且具有P2P能力适用）

3.8.1 注册P2P接口

接口原型

```
/* GO信息 */
typedef struct {
    unsigned char ssid[MAX_LEN_P2P_SSID]; /* 长度为33，包含字符串结束符 */
    unsigned char authMode; /* 1表示wps_wpa，其他值待扩展 */
    unsigned char mac[MAX_LEN_P2P_MAC];
    unsigned char passphrase[MAX_LEN_P2P_PASSWORD]; /* 长度为65，包含字符串结束符 */
    unsigned char ip[MAX_LEN_P2P_IP];
    int freqBand;
    unsigned char channel;
} MagicLinkGoConfig;

typedef struct {
    /* P2P初始化接口，返回值为P2P网口的固定不变的BSSID */
    const char *(*p2pInit)(void);
    /* P2P反初始化接口，将会退出当前P2P模式 */
    void (*p2pDeinit)(void);
    /*
     * 创建GO
     * 创建GO的ssid需要满足设定格式DIRECT-xx-xxxx，xx为随机英文字母，xxxx为设备名可包含空格，
例如DIRECT-wT-xxxx
```

```
     * goIp，入参， GO的IP，将GO配置为此IP
     * frequBand，入参，GO的频段，不支持的此频段创建支持的频段，并通过goCfg返回，
     *            frequBand参数，如支持2.4Gz此处传参为2，如支持5Ghz此处传参为5
     * channelList，入参，GO的信道，从此列表选择创建的信道
     * channelNum，入参，GO信道的数目
     * goCfg，出参，成功创建GO后的配置信息
     */
    int (*creatGo)(const char *goIp, int freqBand, const int *channelList, int
channelNum,
        MagicLinkGoConfig *goCfg);
    int (*removeGo)(void);
    /* 本端作为GC，连接GO，并配置本端的IP */
    int (*gcConnect)(const char *goMac, const char *ip);
    /* 本端作为GC， 断开和GO的连接 */
    int (*gcDisconnect)(const char *goMac);
} P2PAdapterFunc;

/* 注册P2P回调函数 */
int MagicLinkRegP2PFunc(const P2PAdapterFunc *func);
```

入参说明

参数	类型	必选/可选	说明
func	const P2PAdapterFunc *func	必选	P2P适配的函数指针列表

出参说明

```
无
```

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

实现P2PAdapterFunc函数指针对应的函数接口，通过该接口注册。

3.8.2 P2P初始化

接口原型

```
/* P2P初始化 */
const char *(*p2pInit)(void);
```

入参说明

无

出参说明

P2P网口固定不变的BSSID

返回说明

类型	取值	说明
char *	BSSID	成功
	空指针	失败

无

使用说明

调用此接口初始化P2P模块，并将返回值作为P2P网口标识符

3.8.3 P2P去初始化

接口原型

```
/* P2P反初始化接口，将会退出当前P2P模式 */
void (*p2pDeinit)(void);
```

入参说明

无

出参说明

无

返回说明

无

使用说明

调用此接口去初始化设备P2P模式，已有GO则删除GO，已连接GO则断开连接，退出P2P模式。

3.8.4 创建GO

接口原型

```
/* 创建GO热点 */
int (*createGo)(const char *goIp, int freqBand, const int *channelList, int
channelNum, MagicLinkGoConfig *goCfg);
```

入参说明

参数	类型	必选/可选	说明
goIp	const char *	必选	GO的IP，将GO配置为此IP
freqBand	int	必选	GO的频段
channelList	const int *	可选	可选择的GO信道列表
channelNum	int	可选	可选择的GO信道数量

出参说明

参数	类型	必选/可选	说明
goCfg	MagicLinkGoConfig *	必选	成功创建GO后的配置信息

返回说明

类型	取值	说明
int	0	成功
	其他	失败

无

使用说明

要求在Sta模式可调用该接口创建GO热点。
1）调用此接口设备创建GO热点，要求创建GO的ssid格式符合要求，设定格式DIRECT-xx-产品型号，xx为随机两个英文字母，例如DIRECT-wT-Honor70Pro。
2）若freqBand为MAGICLINK_P2P_FREQ_2_4G(2)则创建2.4G的频段热点，若freqBand为MAGICLINK_P2P_FREQ_5G(5)则创建5G的频段热点。
3）channelList为对端所支持的信道列表，即创建的Go热点可选择在此信道中。
4）连接GO成功之后，开发者需要将goCfg完善真实数据后作为出参返回。

3.8.5 关闭GO

接口原型

```
/* 关闭GO */
int (*removeGo)();
```

入参说明

无

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

无

使用说明

在GO模式下调用该接口关闭创建的GO热点

3.8.6 作为GC连接GO

接口原型

```
/* 本端作为GC，连接GO，并配置本端的IP */
int (*gcConnect)(const char *goMac, const char *ip);
```

入参说明

参数	类型	必选/可选	说明
goMac	const char *	必选	对端GO mac
ip	const char *	必选	配置GC的IP

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

无

使用说明

1）本端作为GC时调用此接口设备连接对端GO。要求在连接GO失败时的各种异常场景返回失败错误码。例如：连接超时，GO信息错误等异常场景。

2）连接GO成功需要配置本端IP，根据入参IP配置。

3.8.7 作为GC断开与GO的连接

接口原型

```
/* 本端作为GC， 断开和GO的连接 */
int (*gcDisconnect)(const char *goMac);
```

入参说明

参数	类型	必选/可选	说明
goMac	const char *	必选	对端GOmac

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

作为GC时调用此接口设备断连对端的GO。

3.9 安全相关

初始化前完成注册回调。

3.9.1 获取PIN码

PIN码用于初次使用荣耀智慧空间APP添加设备时确认设备，和设备建立安全通道，HONOR Connect IoT SDK从该接口获取设备PIN码，用于和用户在智慧空间APP上输入的PIN码进行认证确认。

接口原型

```
int MagicLinkRegGetPin(int (*getPin)(char *pinBuf, unsigned int pinBufLen));
```

入参说明

参数	类型	必选/可选	说明
getPin	int (*)(char *, unsigned int)	必选	pinBuf，HONOR Connect IoT SDK存Pin码的地址； pinBufLen，pinBuf长度； 返回值，pin码的长度。

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

1）开发者根据设备模型文件（Profile.json）中pinMode字段，决定是否注册此接口，如果使用输入PIN码、扫码方式获取PIN码，需要注册此接口；使用默认PIN码无需注册此接口；
2）PIN码可以是数字+字母字符串，推荐使用6位纯数字。

3.9.2 软license

License为荣耀设备云颁发，需要通过产线预置到设备中，一台设备一个license.

接口原型

```
int MagicLinkRegGetLicense(int (*getLicense)(char *keyBuf, unsigned int keyBufLen,
char *secretBuf, unsigned int secretBufLen));
```

入参说明

参数	类型	必选/可选	说明
getLicense	int (*)(char *, unsigned int , char *, unsigned int)	必选	keyBuf和KeyBufLen，license的标识的buffer；secretBuf和secretBufLen，license的密钥；

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

1）开发者根据设备模型文件（Profile.json）中authMode字段，决定是否注册此接口，仅在authMode为1时需要注册该接口。
2）设备注册阶段HONOR Connect IoT SDK通过该接口获取软license，一个设备一个license，并在产线预置。
3）软license是设备云认证产品的凭证，请加密存储以防泄露。

3.9.3 产品密钥

产品秘钥为荣耀设备云颁发，需要通过产线预置到设备固件中，一款产品一个密钥。

接口原型

```
int MagicLinkRegGetPrdKey(int (*getPrdKey)(char *keyBuf, unsigned int keyBufLen,
char *secretBuf, unsigned int secretBufLen));
```

入参说明

参数	类型	必选/可选	说明
getPrdKey	int (*)(char *, unsigned int , char *, unsigned int)	必选	keyBuf和KeyBufLen，产品秘钥标识的buffer；secretBuf和secretBufLen，产品秘钥的密钥；

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

1）开发者根据设备模型文件（Profile.json）中authMode字段，决定是否注册此接口，仅在authMode为2时需要注册该接口。
2）设备注册阶段HONOR Connect IoT SDK通过该接口获取产品密钥，一款产品一个密钥，保存在固件中。
3）产品密钥是设备云认证产品的凭证，请加密存储以防泄露。

3.9.4 硬件根密钥

硬件根秘钥是HONOR Connect IoT SDK用于产生密钥的根秘钥，集成方根据产品硬件能力，返回硬件根秘钥。如果没有不注册此接口.

接口原型

```
int MagicLinkRegHardwareRootKey(int (*getHardwareRootKey)(unsigned char *rootKey,
unsigned int rootKeyBufLen));
```

入参说明

参数	类型	必选/可选	说明
getHardwareRootKey	int (*)(char**, char **)	必选	rootKey：HONOR Connect IoT SDK接收硬件根秘钥的buf； rootKeyBufLen：rootKeyBuf的长度； 返回硬件根秘钥的长度

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

接口说明

硬件根秘钥长度不小于32字节。

3.10 HONOR Connect IoT SDK运行参数

3.10.1 运行参数说明

```
/* HONOR Connect IoT SDK运行参数 */
enum MagicLinkAttr {
    /* 配网参数 */
    MAGICLINK_ATTR_NETCFG_TIMEOUT = 0, /* 整数，单位为min，0表示配网永远不超时自动退出 */

    /* 传输参数 */
    MAGICLINK_ATTR_MQTT_BUF_SIZE = 100, /* 整数，为MQTT发送报文长度，如不设置默认为MAGICLINK_ATTR_MQTT_BUF_SIZE(2048)，可设置最大长度为32Kb */
    MAGICLINK_ATTR_CLOUD_HEARTBEAT_INTERVAL = 101, /* 整数，单位为秒，设备和设备云的心跳间隔，最小值：60s，最大值：300s，默认值：60s */

    /* OTA适配使用 */
    MAGICLINK_ATTR_OTA_REBOOT_METHOD = 200, /* 整数，0表示软重启，1表示硬重启 */
    MAGICLINK_ATTR_OTA_FS_PKG_PATH, /* 仅linux使用，存储OTA版本包的路径 */
    MAGICLINK_ATTR_OTA_FS_INS_SCRIPT_PATH, /* 仅linux使用，存储版本包安装脚本的路径 */
    MAGICLINK_ATTR_OTA_FLASH_MASTER_PKG_ADDR, /* 整数，固件主分区flash起始地址，必须为4K对齐 */
    MAGICLINK_ATTR_OTA_FLASH_SLAVE_PKG_ADDR, /* 整数，固件备份分区flash起始地址，必须为4K对齐 */
    MAGICLINK_ATTR_OTA_FLASH_MASTER_PKG_SIZE, /* 整数，OTA固件主分区flash存储空间大小，单位为字节，必须为4K的整数倍，建议该值设置要比实际固件大，考虑升级可扩展性 */
    MAGICLINK_ATTR_OTA_FLASH_SLAVE_PKG_SIZE, /* 整数，OTA固件备份分区flash存储空间大小，单位为字节，必须为4K的整数倍，建议该值设置要比实际固件大，考虑升级可扩展性 */
    MAGICLINK_ATTR_OTA_MCU_TRANS_TIMEOUT, /* 整数，模组给MCU发送MCU升级数据包的超时时间，单位为秒，默认值为30，建议实测发送需要的时间 */
```

```
    MAGICLINK_ATTR_OTA_MCU_INSTALL_TIMEOUT, /* 整数，MCU安装重启超时时长，单位为秒，默认值为30 */
    MAGICLINK_ATTR_OTA_DOWNLOAD_SINGLE_PKG_LEN, /* 整数，OTA分包下载单包长度，仅Linux系统设备使用，单位为KB，取值范围为4到512，默认值为64 */
    MAGICLINK_ATTR_OTA_MODULE_INSTALL_TIMEOUT, /* 整数，MODULE安装重启的超时时间，单位为秒，默认值为30，取值范围为30到300 */

    /* 配置保存适配使用 */
    MAGICLINK_ATTR_CONFIG_PATH = 300, /* 仅linux使用，存储SDK数据的路径 */
    MAGICLINK_ATTR_CONFIG_BAK_PATH, /* 仅linux使用，存储SDK数据的备份路径 */
    MAGICLINK_ATTR_CONFIG_FLASH_BASE_ADDR, /* flash使用，48KB存储SDK数据起始地址 */
    MAGICLINK_ATTR_READONLY_CONFIG_PATH, /* 只读配置文件的存储地址，需设置为只读分区的路径 */

    /* unsigned int类型，接收到setProperty请求后自动上报的模式；0：上报setProperty下发属性所属服务下的所有属性状态；1：上报setProperty下发属性的状态；默认值：0 */
    MAGICLINK_ATTR_AUTO_REPORT_MODE = 400,

    /* 本地化信息，需要在设备已经完成云端注册时才能获取;设备注册成功后，修改手机侧时区等信息无法同步至设备。 */
    MAGICLINK_ATTR_COUNTRYCODE = 450, /* char *类型，大小不超过4字节，参照ISO 3166-1 alpha-2 国家码，如中国国家码为"CN" */
    MAGICLINK_ATTR_LANGUAGECODE, /* char *类型，大小不超过16字节，参照ISO 639-1 语言码，如汉语的语言码是"zh" */
    MAGICLINK_ATTR_TIMEZONE, /* int 类型，参照UTC时间时区划分，如北京的时区是8 */

    /* DFX日志收集使用 */
    MAGICLINK_ATTR_LOG_OUTPUT_PATH = 600, /* 字符类型，SDK日志落盘路径和日志打包文件保存路径。如果需要收集开发者的日志，开发者需要在这个路径中同时预留足够空间。 */
    MAGICLINK_ATTR_DFX_LOG_FILE_SIZE, /* 整数，SDK单个日志文件大小，单位为M。取值范围为1M到8M，默认值为1M */
    MAGICLINK_ATTR_DFX_LOG_FILE_NUM, /* 整数，SDK日志文件最大数量，取值范围为1-50，默认值为10 */
    MAGICLINK_ATTR_DFX_EXT_LOG_PKG_TIME, /* 整数，单位为秒，默认时间为30s，厂商日志收集的最长时间。
    SDK通知厂商进行日志收集后等待日志收集完成通知，如果等待超时，SDK不收集厂商日志，仅收集SDK日志 */
    MAGICLINK_ATTR_CLOUD_PROXY = 700, /* 仅设备支持蓝牙代理上网时使用，使用代理方式注册到设备云。取值范围0到1，默认值为0 */
    MAGICLINK_ATTR_BUTT,
};
```

3.10.2 设置运行参数

接口原型

```
int MagicLinkSetAttr(enum MagicLinkAttr attr, void *value, unsigned int valueLen);
```

入参说明

参数	类型	必选/可选	说明
attr	enum MagicLinkAttr	必选	HONOR Connect IoT SDK属性，参考MagicLinkAttr定义（超链接跳转到上面）
value	void *	必选	属性值
valueLen	unsigned int	必选	属性值长度

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

接口说明

无。

3.10.3 查询运行参数

接口原型

```
int MagicLinkGetAttr(enum MagicLinkAttr attr, void *buff, unsigned int buffLen,
unsigned int *outLen);
```

入参说明

参数	类型	必选/可选	说明
attr	enum MagicLinkAttr	必选	HONOR Connect IoT SDK属性，参考MagicLinkAttr定义（超链接跳转到上面）
value	void *	必选	属性值buff
valueLen	unsigned int	必选	属性值buff长度

出参说明

参数	类型	必选/可选	说明
outLen	unsigned int *	必选	属性值长度

返回说明

类型	取值	说明
int	0	成功
	其他	失败

接口说明

无。

3.11 日志接口

3.11.1 注册日志接口

接口原型

```
enum MagicLinkLogLevel {
    LOG_LEVEL_DEBUG,
    LOG_LEVEL_INFO,
    LOG_LEVEL_WARN,
    LOG_LEVEL_ERR,
    LOG_LEVEL_CRITICAL,
    LOG_LEVEL_DATA,
    LOG_LEVEL_BUTT
};

int MagicLinkRegLogFunc(void (*logFunc)(const char *func, int line, enum
MagicLinkLogLevel level,
    const char *format, ...));
```

入参说明

参数	类型	必选/可选	说明
logFunc	日志接口函数	必选	日志打印接口

出参说明

```
无
```

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

开发者若需自定义日志打印功能，通过该接口注册。

3.11.2 日志接口

接口原型

```
void (*logFunc)(const char *func, int line, enum MagicLinkLogLevel level, const
char *format, ...);
```

入参说明

参数	类型	必选/可选	说明
func	const char *	必选	日志所在函数的函数名
line	int	必选	日志所在文件的行号
level	enum MagicLinkLogLevel	必选	日志等级
format	const char *	必选	日志格式化字符串
...	可变参数	可选	与format配合使用

出参说明

```
无
```

返回说明

```
无
```

使用说明

日志输出接口

3.11.3 日志级别设置

接口原型

```
int MagicLinkSetLogLevel(enum MagicLinkLogLevel level);
```

入参说明

参数	类型	必选/可选	说明
level	enum MagicLinkLogLevel	必选	日志级别

出参说明

```
无
```

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

开发者需调整日志输出级别时，调用该接口。例，设置LOG_LEVEL_ERR，将输出日志级别在LOG_LEVEL_ERR到LOG_LEVEL_DATA之间的日志

3.11.4 日志级别获取

接口原型

```
enum MagicLinkLogLevel MagicLinkGetLogLevel(void);
```

入参说明

```
无
```

出参说明

```
无
```

返回说明

类型	取值	说明
enum MagicLinkLogLevel	整数	HONOR Connect IoT SDK当前使用的日志级别

使用说明

获取当前HONOR Connect IoT SDK使用的日志级别

3.12 查询HONOR Connect IoT SDK版本号

接口原型

```
int MagicLinkGetSDKVersion(char *verBuf, unsigned int verBufLen);
```

入参说明

参数	类型	必选/可选	说明
verBuf	char *	必选	HONOR Connect IoT SDK版本号buffer
verBufLen	unsigned int	必选	HONOR Connect IoT SDK版本号buffer长度（建议此buffer长度为32字节）

出参说明

参数	类型	必选/可选	说明
verBuf	char *	必选	函数执行成功后，HONOR Connect IoT SDK版本号将写入此buffer（带字符串结束符）

返回说明

类型	取值	说明
int	0	成功
	其他	失败

接口说明

无。

3.13 时间查询接口

3.13.1 获取本地时间

接口原型

```
int MagicLinkGetLocalTime(struct tm *time)
```

入参说明

参数	类型	必选/可选	说明
time	struct tm *	必选	标准库函数time.h中表示时间的结构体

出参说明

无。

返回说明

类型	取值	说明
int	0	成功
	-1	magiclink获取时间失败
	-99	输入参数错误
	-101	本地时间未与云端时间同步

接口说明

获取当前设备的本地时间。

3.13.2 获取UTC时间

接口原型

```
int MagicLinkGetUTCTime(struct tm *time)
```

入参说明

参数	类型	必选/可选	说明
time	struct tm *	必选	标准库函数time.h中表示时间的结构体

出参说明

无。

返回说明

类型	取值	说明
int	0	成功
	-1	magiclink获取时间失败
	-99	输入参数错误
	-101	本地时间未与云端时间同步

接口说明

获取当前设备的UTC时间。

3.14 设备device ID查询接口

接口原型

```
int MagicLinkGetDeviceId(char *devIdBuf, unsigned int devIdBufLen);
```

入参说明

参数	类型	必选/可选	说明
devIdBuf	char *	必选	设备device ID buffer
devIdBufLen	unsigned int	必选	设备device ID buffer的长度,长度不小于48

出参说明

参数	类型	必选/可选	说明
devIdBuf	char *	必选	函数执行成功后，设备device ID将写入此buffer

返回说明

类型	取值	说明
int	0	成功
	其他	失败

接口说明

获取当前设备的device ID。需要在设备已经完成云端注册时才能通过该接口获取到device ID，设备未注册之前调用该接口将返回失败。

3.15 云端环境配置

3.15.1 设置云端环境

接口原型

```
int MagicLinkSetCloudEnv(int cloudEnv)
```

入参说明

参数	类型	必选/可选	说明
cloudEnv	int	必选	设置云端环境：1-生产环境 2-预生产环境 3-测试环境 4-开发环境

出参说明

无。

返回说明

类型	取值	说明
int	0	成功
	其他	失败

接口说明

设置云端环境，供使用自注册功能的设备使用。
注意：设置云端环境之后，需要同步更新设备的固定秘钥或者License为对应云环境颁发的固定秘钥或者License。

3.15.2 获取云端环境

接口原型

```
int MagicLinkGetCloudEnv(void)
```

入参说明

```
无。
```

出参说明

无。

返回说明

类型	取值	说明
int	1	生产环境
	2	预生产环境
	3	测试环境
	4	开发环境
	其他值	无效值

接口说明

查询云端环境，供使用自注册功能的设备使用。

3.16 注册接收设备绑定时间回调（仅linux系统适用）

开发者可以注册接收设备绑定时间回调给SDK，SDK在设备注册前调用回调接口设置设备绑定时间到系统时间，从而优化设备配网注册上线时间性能。该接口为可选实现，如开发者不实现该接口，则SDK的配网注册上线时间会受到系统NTP同步时间快慢的影响。

接口原型

```
int MagicLinkRegRecvDevBindTime(int (*recvDevBindTime)(long long localTime));
```

入参说明

参数	类型	必选/可选	说明
recvDevBindTime	int (*)(long long localTime)	必选	localTime为本地时间，精度为秒，开发者可以使用系统接口设置时间。

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

1. 该接口设置的时间为从手机同步过来的系统时间，不能保证该时间与NTP时间的完全同步，该时间仅用于设备注册到云端时进行认证校验。当设备联网成功之后，开发者需要重新进行时间同步，保证系统时间与NTP时间同步。
2. 通常设置系统时间需要root权限，开发者在实现系统时间设置接口时，需要保证有设置系统时间的权限。
3. 如果在该接口调用时，设备已完成从NTP同步时间，则无需重复设置设备绑定时间到系统时间。

3.17 注册获取本地化信息回调

开发者可以注册获取本地化信息回调给SDK，SDK调用回调接口从开发者获取国家码、语言码和时区信息。

接口原型

```
int MagicLinkRegGetLocale(int (*getLocale)(char *country, unsigned int
countryBufLen, char *lang, unsigned int langBufLen, int *timeZone));
```

入参说明

参数	类型	必选/可选	说明
getLocale	int (*getLocale)(char *country, unsigned int countryBufLen, char *lang, unsigned int langBufLen, int *timeZone)	必选	国家码最大长度5字节，语言码最大长度17字节

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

国家码内容参照ISO 3166-1 alpha-2，如中国国家码为"CN"；
语言码参照ISO 639-1 语言码，如汉语的语言码是"zh"。
时区为UTC时间时区，如北京的时区是8。

3.18 日志收集开发（仅linux系统适用，仅中国国内适用）

开发者如果需要SDK在收集日志时，同时收集开发者的日志，则需要注册日志收集回调。开发者将需要上传的日志打包到一个文件中，并在打包完成后通知SDK。

3.18.1 注册日志收集回调

接口原型

```
int MagicLinkRegLogCollect(int (*logCollect)());
```

入参说明

参数	类型	必选/可选	说明
logCollect	int (*)()	必选	

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

```
1. 该回调接口为异步接口，开发者需要在开始日志打包以后返回。
```

3.18.2 通知SDK日志收集完成

接口原型

```
int MagicLinkNotifyLogCollect(int status, const char *fileFullName, unsigned int
nameLen);
```

入参说明

参数	类型	必选/可选	说明
status	int	必选	日志收集成功传入0，日志收集失败传入-1。
fileFullName	char *	必选	打包日志完整绝对路径，包括文件路径和文件名。
nameLen	int	必选	打包日志完整绝对路径长度。

出参说明

无

返回说明

类型	取值	说明
int	0	成功
	其他	失败

使用说明

1. 开发者需要保证SDK有打包日志文件的读写权限，SDK在完成日志上传以后，会删除fullName参数对应的日志文件。
2. 开发者需要在MAGICLINK_ATTR_DFX_EXT_LOG_PKG_TIME参数定义的时间内完成日志收集并通知SDK日志收集完成和日志打包路径。

3.19 设备云透传命令字

开发者如果需要使用约定的命令字向设备云发送业务请求，需要通过SDK提供的接口实现请求发送和响应处理。

3.19.1 设备云响应处理回调注册

接口原型

```
int MagicLinkRegCloudAPICb(const char *cmd, int (*cb)(const char *errorCode, const char *cmd, const char *payload));
```

入参说明

参数	类型	必选/可选	说明
cmd	const char *	必选	响应命令字
cb	int (*)(const char *errorCode, const char *cmd, const char *payload)	必选	注册的响应回调函数

出参说明

无

返回说明

类型	取值	说明
int	0	成功
int	-904001	命令字不支持
int	-1	其他类型的失败

使用说明

1. 开发者应在MagicLinkInit()接口调用成功后，使用该接口注册响应处理回调。

3.19.2 发送设备云透传命令请求

接口原型

```
int MagicLinkSendAPI2Cloud(const char *cmd, const char *payload);
```

入参说明

参数	类型	必选/可选	说明
cmd	const char *	必选	请求命令字
payload	const char *	必选	请求内容

出参说明

无

返回说明

类型	取值	说明
int	0	成功
int	-904001	命令字不支持
int	-1	其他类型的失败

使用说明

1. 若在发出请求10s后，没有收到响应，可以当作超时处理。

3.19.3 SDK支持命令字

命令字类型	请求命令字	响应命令字
天气查询	API_WEATHER_REQ	API_WEATHER_RSP
历史记录上报	API_HIS_DATA_REPORT_REQ	API_HIS_DATA_REPORT_RSP

3.19.4 请求/响应内容样例

天气查询请求

```
{
    "latitude": "34.149",                          // 纬度，3位小数
    "longitude": "108.993",                        // 经度，3位小数
    "coordinate": 2,                               // 1：GCJ-02 （国内标准）2：WGS-84
（国外使用）
    "areaRange": 1,                                // 1：区级；  2：市级
    "queryWeatherType": [1, 2],                    // 1：当天实况天气  2：预报天气
    "forecastDayRange": 1                          // 预测天数，取值范围：[1, 10]；   以当
天为第1天  queryWeatherType包含2时必填。
}
```

天气查询响应

```
{
    "cityName": "长安区",                          // 定位名称（区or市）
    "weather": {                                   // 当前实况天气信息queryWeatherType =
1 时涉及
        "weatherCode": 2,                          // 天气类型编码  参考天气信息编码表
        "weatherDesc": "阴",                       // 天气文字描述
        "currentTemperature": "26.0",              // 当前温度  摄氏度，1位小数
        "relativeHumidity": "86",                  // 相对湿度
        "windCode": 8,                             // 风向类型编码  参考风向信息编码表
        "windDirectionDesc": "西北",               // 风向
        "windPower": 2,                            // 风力，单位：级  参考风力等级表
        "weatherIndexes": [{                       // 天气相关指数
            "weatherKey": "UVIndex",               // 天气相关指数标记  参考天气指数标记表
            "weatherValue": "1"                    // 天气相关指数具体值  各指数标准规范不
同，统一返回字符串类型
        }],
        "updateTime": 1690447947000                // 发布时间
    },
    "weatherDayForecasts": [{                      // 天气预报信息，以天为单位，
queryWeatherType = 2 时涉及
        "minTemperature": "21.0",                  // 最小温度  摄氏度，1位小数
        "maxTemperature": "32.0",                  // 最高温度  摄氏度，1位小数
        "dayTimeWeather": {                        // 白天天气预报
            "weatherCode": 8,                      // 天气类型编码  参考天气信息编码表
            "weatherDesc": "中雨",                 // 天气文字描述
            "windCode": 6,                         // 风向类型编码  参考风向信息编码表
            "windDirectionDesc": "西南",           // 风向
            "windPower": 1                         // 风力，单位：级  参考风力等级表
        },
        "nightTimeWeather": {                      // 夜间天气预报
            "weatherCode": 8,
            "weatherDesc": "中雨",
            "windCode": 6,
            "windDirectionDesc": "西南",
            "windPower": 1
        }
    }]
}
```

历史记录上报请求

以扫地机清扫记录上报为例，以下样例表示在"1705026371000"时间点上报了一条记录：

```
{
    "tableName" : "QEAU_EVENT_CLEAN_RECORD",     // 事件表名称，string，QEAU为产品ID
    "data" : {
        "timestamp":[1705026371000],             // 首次上报时间，重传请求时需要保持一致，
array<long>，UTC时间戳，精确到毫秒
        "columns":[
            [1],                                 // 地图id，array<int>
            [50],                                // 清扫面积，array<int>，单位为平方米
            [60],                                // 清扫时间，array<int>，单位为秒
            [1705026371000],                     // 结束时间，array<long>， UTC时间戳，
精确到毫秒
            [0],                                 // 清扫状态，array<int>
            [0],                                 // 清扫模式，array<int>
            [0]                                  // 错误码，array<int>
        ]
    }
}
```

历史记录上报响应

以扫地机清扫记录响应为例，以下样例表示清扫记录上报成功：

```
{
    "errorCode": "0",
    "errorDesc": "success"
}
```