物联网开发平台

客户端 API

腾讯云

文档目录

客户端 API
设备端

最近更新时间：2026-07-30 12:00:36

概述

本文档描述设备端公开 C API。接口线程安全（可任意线程调用），异步回调契约为：同步返回成功则一定回调，同步返回失败则一定不回调（个别接口另有说明）。推荐生命周期：tc_iot_init → tc_iot_login → 各业务模块 init → 业务运行 → 各业务模块 deinit → tc_iot_deinit 。
SDK 模块一览：

模块	说明
核心生命周期	核心生命周期（连接与设备身份）
物模型	物模型（属性 / 事件 / 行为）
音视频	音视频通话 / IPC 监控
AI 对话	AI 对话（AITalk）
类型定义	音视频公共类型（AV / AITalk 共用）
错误码	SDK 统一错误码

API 索引：

核心生命周期（tc_iot.h）

函数列表	描述
tc_iot_init	初始化 SDK 核心（创建 iot 消息循环并异步完成内部模块初始化）
tc_iot_deinit	反初始化 SDK；阻塞等待内部清理完成后返回
tc_iot_login	使用三元组登录（异步 MQTT 连接，内部可重试）
tc_iot_logout	登出并清除本地凭证（异步，无独立完成回调）
tc_iot_dynamic_register	动态注册设备（同一时刻仅允许一路）
tc_iot_get_ntp_time	读取 NTP 同步后的 UTC 毫秒时间戳

物模型（tc_iot_data_model.h）

函数列表	描述
tc_iot_data_model_init	初始化物模型并订阅下行 topic
tc_iot_data_model_report_property	上报属性
tc_iot_data_model_report_event	上报事件
tc_iot_data_model_deinit	反初始化；同步清除单例后异步退订

音视频（tc_iot_av.h）

函数列表	描述
tc_iot_av_init	初始化 AV 模块并注册观察者（内部拷贝 observer）
tc_iot_av_deinit	反初始化 AV；同步清除单例后异步退订信令
tc_iot_av_get_contacts	拉取联系人列表
tc_iot_av_call	发起呼出
tc_iot_av_accept	接听呼入；仅在振铃态有效，否则回调 AV_ON_*
tc_iot_av_reject	拒绝呼入；仅在振铃态有效
tc_iot_av_hangup	挂断当前会话；无对应会话时回调 TC_IOT_ERR_AV_ON_IDLE
tc_iot_create_video_channel	创建视频推流通道
tc_iot_create_audio_channel	创建全局唯一音频推流通道
tc_iot_destroy_video_channel	销毁视频通道
tc_iot_destroy_audio_channel	销毁音频通道
tc_iot_push_video_frame	推送一帧视频（调用方线程同步）
tc_iot_push_audio_frame	推送一帧音频（调用方线程同步）

AI 对话（tc_iot_aitalk.h）

函数列表	描述
tc_iot_aitalk_init	初始化 AITalk
tc_iot_aitalk_deinit	反初始化；阻塞等待内部 teardown 完成
tc_iot_aitalk_start_speak	开始对话
tc_iot_aitalk_stop_speak	停止对话（异步投递，空闲时幂等成功）
tc_iot_aitalk_send_audio	发送一帧上行音频
tc_iot_aitalk_send_text	发送文本给 LLM
tc_iot_aitalk_register_contacts	注册可供机器人呼叫的联系人
tc_iot_aitalk_interrupt	打断机器人当前播报（连续模式下有效）

核心生命周期（tc_iot.h）

核心生命周期（连接与设备身份）

线程模型

本头文件接口可在任意线程调用；内部序列化到名为 "iot" 的消息循环执行。

回调 / 通知模型

- on_log：日志回调线程由内部日志模块决定，可能与调用方线程不同。
- on_device_event：在 iot 消息循环侧触发；event_data 当前恒为 NULL； 回调内勿做耗时操作。ONLINE 仅表示 MQTT 已连，不携带校时， 壁钟请用 tc_iot_get_ntp_time（未就绪返回 NTP_TIME_NOT_READY）。
- tc_iot_login_cb / tc_iot_dynamic_register_cb： 同步返回 SUCCESS 则一定会回调；同步失败则一定不回调。

生命周期

推荐顺序：tc_iot_init → tc_iot_login（等待回调成功）→ 各业务模块 init → 业务运行 → 各业务模块 deinit → tc_iot_deinit。 tc_iot_logout 可选；deinit 会断开 MQTT。

业务约束

- 重复 init 返回 ALREADY_INITIALIZED；未 init 调用业务接口返回 NOT_INITIALIZED。
- tc_iot_init 同步 SUCCESS 仅表示任务已投递；异步初始化失败时后续 login 会以回调失败体现。

- tc_iot_deinit 为阻塞接口；动态注册进行中时拒绝 deinit（DYNAMIC_REGISTER_BUSY）。
- 动态注册成功后的 device_key 须由调用方写入非易失存储， 再作为 device_secret 登录。

函数列表	描述
tc_iot_init	初始化 SDK 核心（创建 iot 消息循环并异步完成内部模块初始化）
tc_iot_deinit	反初始化 SDK；阻塞等待内部清理完成后返回
tc_iot_login	使用三元组登录（异步 MQTT 连接，内部可重试）
tc_iot_logout	登出并清除本地凭证（异步，无独立完成回调）
tc_iot_dynamic_register	动态注册设备（同一时刻仅允许一路）
tc_iot_get_ntp_time	读取 NTP 同步后的 UTC 毫秒时间戳

tc_iot_init

tc_iot_init

tc_iot_error_e tc_iot_init (const tc_iot_config_s *config)

初始化 SDK 核心（创建 iot 消息循环并异步完成内部模块初始化）

参数	描述
config	初始化配置，不可为 NULL

返回值	描述
TC_IOT_ERR_SUCCESS	任务已投递
TC_IOT_ERR_INVALID_ARGUMENT	参数非法
TC_IOT_ERR_ALREADY_INITIALIZED	已初始化
TC_IOT_ERR_OUT_OF_MEMORY	内存不足

tc_iot_deinit

tc_iot_deinit

tc_iot_error_e tc_iot_deinit (void)

反初始化 SDK；阻塞等待内部清理完成后返回

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_NOT_INITIALIZED	未初始化
TC_IOT_ERR_DYNAMIC_REGISTER_BUSY	动态注册进行中
TC_IOT_ERR_OUT_OF_MEMORY	内存不足

tc_iot_login

tc_iot_login

tc_iot_error_e tc_iot_login (const tc_iot_device_info_s *device_info, tc_iot_login_cb c

使用三元组登录（异步 MQTT 连接，内部可重试）

同步返回 SUCCESS 仅表示任务已投递；最终结果以 `callback` 为准。 已连接时回调 TC_IOT_ERR_ALREADY_CONNECTED 。

参数	描述
device_info	设备三元组；product_id/device_id/device_secret 必填
callback	登录结果回调；同步 SUCCESS 则一定回调，失败则不回调

返回值	描述
TC_IOT_ERR_SUCCESS	任务已投递
TC_IOT_ERR_INVALID_AR	参数非法

GUMENT	
TC_IOT_ERR_NOT_INITIALIZED	未 init
TC_IOT_ERR_OUT_OF_MEMORY	内存不足

tc_iot_logout

tc_iot_logout

tc_iot_error_e tc_iot_logout (void)

登出并清除本地凭证（异步，无独立完成回调）

进行中的 login 会被取消并以 TC_IOT_ERR_NOT_CONNECTED 回调。

返回值	描述
TC_IOT_ERR_SUCCESS	任务已投递
TC_IOT_ERR_NOT_INITIALIZED	未 init
TC_IOT_ERR_OUT_OF_MEMORY	内存不足

tc_iot_dynamic_register

tc_iot_dynamic_register

tc_iot_error_e tc_iot_dynamic_register (const tc_iot_dynamic_register_param_s *par
tc_iot_dynamic_register_cb callback)

动态注册设备（同一时刻仅允许一路）

成功后用回调中的 device_key 作为 device_secret 登录。

参数	描述
param	注册参数

callback	注册结果回调；同步 SUCCESS 则一定回调，失败则不回调

返回值	描述
TC_IOT_ERR_SUCCESS	任务已投递
TC_IOT_ERR_INVALID_ARGUMENT	参数非法
TC_IOT_ERR_NOT_INITIALIZED	未 init
TC_IOT_ERR_DYNAMIC_REGISTER_BUSY	已有注册进行中
TC_IOT_ERR_OUT_OF_MEMORY	内存不足

tc_iot_get_ntp_time

tc_iot_get_ntp_time

tc_iot_error_e tc_iot_get_ntp_time (uint64_t *utc_time_ms)

读取 NTP 同步后的 UTC 毫秒时间戳

MQTT ONLINE 后异步触发同步；未完成前返回 NTP_TIME_NOT_READY。

参数	描述
utc_time_ms	输出 UTC 毫秒时间戳，不可为 NULL

返回值	描述
TC_IOT_ERR_SUCCESS	已同步并写入
TC_IOT_ERR_INVALID_ARGUMENT	参数非法
TC_IOT_ERR_NTP_TIME_NOT_READY	尚未同步完成

tc_iot_log_level_e

tc_iot_log_level_e
日志输出阈值（数值越大输出越详细）。设置为某级别时，输出该级别及更严重的日志（不含更详细级别）。

枚举	取值	描述
TC_IOT_LOG_LEVEL_NONE	0	关闭日志输出
TC_IOT_LOG_LEVEL_ERROR	1	仅输出错误
TC_IOT_LOG_LEVEL_WARN	2	输出警告与错误
TC_IOT_LOG_LEVEL_INFO	3	输出信息、警告与错误
TC_IOT_LOG_LEVEL_DEBUG	4	输出全部日志（含调试，最详细）

tc_iot_device_event_type_e

tc_iot_device_event_type_e
设备在线状态事件（MQTT 连接态映射；不含墙钟时间）

枚举	取值	描述
TC_IOT_DEV_EVENT_ONLINE	0	MQTT 已连接（≠ NTP 已同步）
TC_IOT_DEV_EVENT_OFFLINE	1	MQTT 已断开
TC_IOT_DEV_EVENT_RECONNECTING	2	MQTT 重连中

tc_iot_login_cb

tc_iot_login_cb
登录结果回调

typedef void(* tc_iot_login_cb)(tc_iot_error_e error_code, const char *error_message)

参数	描述
error_code	成功时为 TC_IOT_ERR_SUCCESS
error_message	错误描述，可为 NULL

tc_iot_dynamic_register_cb

tc_iot_dynamic_register_cb
动态注册结果回调

typedef void(* tc_iot_dynamic_register_cb) (tc_iot_error_e error_code, const char *error_message, const char *device_key)

参数	描述
error_code	成功时为 TC_IOT_ERR_SUCCESS
error_message	错误描述，可为 NULL
device_key	成功时为设备密钥；须在回调返回前拷贝并写入非易失存储，再作为后续 login 的 device_secret

物模型（tc_iot_data_model.h）

物模型（属性 / 事件 / 行为）

线程模型
本头文件接口可在任意线程调用；上报与下行处理在 iot 消息循环执行。

回调 / 通知模型
- on_report_*_result_cb：上报结果（含超时）；user_data 为 report 传入值。
- on_receive_property_changed_cb：云端属性控制。
- on_receive_new_action_cb：收到行为调用；调用方填充 action_output_data_* 后返回 0 表示成功回复，非 0 表示失败回复； 回调返回后 action 对象即释放，禁止缓存指针。

生命周期

tc_iot_login 成功 → tc_iot_data_model_init → report_* → tc_iot_data_model_deinit

业务约束

- 标量类型用 data_value 对应字段；STRING 指针在构建 JSON 期间须有效。
- 可一次 report 多条；每条结果经 on_report_*_result_cb 返回。
- OBJECT / ARRAY 枚举已预留，当前实现未支持，请勿使用。

函数列表	描述
tc_iot_data_model_init	初始化物模型并订阅下行 topic
tc_iot_data_model_report_property	上报属性
tc_iot_data_model_report_event	上报事件
tc_iot_data_model_deinit	反初始化；同步清除单例后异步退订

tc_iot_data_model_init

```
tc_iot_error_e tc_iot_data_model_init(tc_iot_data_model_callback_t data_model_ca
```

初始化物模型并订阅下行 topic

须 login 成功。

参数	描述
data_model_callback	回调表

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_ALREADY_INITIALIZED	重复初始化
TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪

TC_IOT_ERR_POST_TASK_FAILED	投递到 iot 消息循环失败
TC_IOT_ERR_FAILURE	通用失败

tc_iot_data_model_report_property

tc_iot_data_model_report_property

tc_iot_error_e tc_iot_data_model_report_property (tc_iot_data_model_property_t *data_
int property_num, void *report_cb_user

上报属性

property_num > 0；同步 SUCCESS 表示发布任务已投递， 逐条结果经 on_report_property_result_cb 返回。

参数	描述
data_model_property	属性数组
property_num	数量
report_cb_user_data	透传到结果回调

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪
TC_IOT_ERR_INVALID_ARGUMENT	参数非法（空指针、非法取值等）
TC_IOT_ERR_OUT_OF_MEMORY	内存不足
TC_IOT_ERR_POST_TASK_FAILED	投递到 iot 消息循环失败
TC_IOT_ERR_FAILURE	通用失败

tc_iot_data_model_report_event

```
tc_iot_error_e tc_iot_data_model_report_event(tc_iot_data_model_event_t *data_mo
tc_iot_data_model_report_event                event_num, void *event_cb_user_data)
```

上报事件

语义与 report_property 相同，结果经 on_report_event_result_cb。

参数	描述
data_model_event	事件数组
event_num	数量
event_cb_user_data	透传到结果回调

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪
TC_IOT_ERR_INVALID_ARGUMENT	参数非法（空指针、非法取值等）
TC_IOT_ERR_OUT_OF_MEMORY	内存不足
TC_IOT_ERR_POST_TASK_FAILED	投递到 iot 消息循环失败
TC_IOT_ERR_FAILURE	通用失败

tc_iot_data_model_deinit

```
tc_iot_error_e tc_iot_data_model_deinit(void)
```

反初始化；同步清除单例后异步退订

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪

data_model_data_type_t

data_model_data_type_t
物模型数据类型

枚举	取值	描述
DATA_MODEL_DATA_TYPE_BOOL	0	布尔
DATA_MODEL_DATA_TYPE_INT	1	整数
DATA_MODEL_DATA_TYPE_FLOAT	2	浮点
DATA_MODEL_DATA_TYPE_STRING	3	字符串
DATA_MODEL_DATA_TYPE_ENUM	4	枚举
DATA_MODEL_DATA_TYPE_TIME	5	时间
DATA_MODEL_DATA_TYPE_OBJECT	6	预留；当前实现未支持
DATA_MODEL_DATA_TYPE_ARRAY	7	预留；当前实现未支持

data_model_event_type_t

data_model_event_type_t
事件类型

枚举	取值	描述
DATA_MODEL_EVENT_TYPE_INFO	0	信息类事件

DATA_MODEL_EVENT_TYPE_ALERT	1	告警类事件
DATA_MODEL_EVENT_TYPE_FAULT	2	故障类事件

data_model_report_result_t

data_model_report_result_t
上报结果

枚举	取值	描述
DATA_MODEL_REPORT_SUCCESS	0	云端接受
DATA_MODEL_REPORT_REJECTED	1	云端拒绝
DATA_MODEL_REPORT_NO_RESPONSE	2	无云端响应
DATA_MODEL_REPORT_LOCAL_TIMEOUT	3	本地等待超时

data_model_bool_t

data_model_bool_t
布尔属性值类型

typedef bool data_model_bool_t

data_model_int_t

data_model_int_t
整型属性值类型

typedef int32_t data_model_int_t

data_model_float_t

data_model_float_t
浮点属性值类型

typedef double data_model_float_t

data_model_string_t

data_model_string_t
字符串属性值类型（调用方持有指针）

typedef char* data_model_string_t

data_model_enum_t

data_model_enum_t
枚举属性值类型

typedef uint32_t data_model_enum_t

data_model_time_t

data_model_time_t
时间属性值类型（通常为 Unix 秒）

typedef uint32_t data_model_time_t

tc_iot_data_model_callback_t

tc_iot_data_model_callback_t
物模型回调表

typedef struct tc_iot_data_model_callback tc_iot_data_model_callback_t

音视频（tc_iot_av.h）

音视频通话 / IPC 监控

线程模型

本头文件接口可在任意线程调用；信令类操作投递到 iot 消息循环。 push_* 在调用方线程同步执行（写入 media_stream）。

回调 / 通知模型

- tc_iot_av_observer_s：init 时拷贝；信令回调在 iot 消息循环触发； 音视频帧回调可能来自 TRTC 媒体线程；所有回调须尽快返回、勿阻塞。
- tc_iot_av_operation_cb：call/accept/reject/hangup 的 MQTT 发布结果； 同步 SUCCESS 则一定回调，同步失败则一定不回调。
- accept/reject/hangup 要求 callback 非 NULL；call 也应提供有效 callback。

生命周期

tc_iot_init + tc_iot_login 成功 → tc_iot_av_init →（可选提前）create_*channel* → *按会话推流 / 呼叫 API* → *destroy* / hangup → tc_iot_av_deinit

推流时机

- 监控：收到 on_monitor_begin 后持续 push_*；on_monitor_end 后可停。多路监控复用同一音频通道，每路视频会话期间仍须送音频。
- VOIP：通话建立后再推（呼出等 on_call_accepted；呼入等 accept 成功）。
- 通道可预先 create；无活跃会话时 push 只写入 media_stream， 是否进房/上云由会话侧决定。

业务约束

- 视频通道槽位最多 4 个，channel_id 须唯一（可为任意正整数）。
- 音频通道全局仅 1 路；已存在时 create 返回 NULL，须复用指针。
- push_audio_frame 仅接受 16-bit PCM；非 PCM 返回 INVALID_ARGUMENT。
- 视频 Annex-B；IDR 须含 SPS/PPS（H265 含 VPS）；不支持 B 帧。
- 监控与 VOIP 互斥由状态机约束（冲突时 AV_ON_*）。
- tc_iot_av_get_contacts 当前为占位实现。

函数列表	描述
tc_iot_av_init	初始化 AV 模块并注册观察者（内部拷贝 observer）
tc_iot_av_deinit	反初始化 AV；同步清除单例后异步退订信令

tc_iot_av_get_contacts	拉取联系人列表
tc_iot_av_call	发起呼出
tc_iot_av_accept	接听呼入；仅在振铃态有效，否则回调 AV_ON_*
tc_iot_av_reject	拒绝呼入；仅在振铃态有效
tc_iot_av_hangup	挂断当前会话；无对应会话时回调 TC_IOT_ERR_AV_ON_IDLE
tc_iot_create_video_channel	创建视频推流通道
tc_iot_create_audio_channel	创建全局唯一音频推流通道
tc_iot_destroy_video_channel	销毁视频通道
tc_iot_destroy_audio_channel	销毁音频通道
tc_iot_push_video_frame	推送一帧视频（调用方线程同步）
tc_iot_push_audio_frame	推送一帧音频（调用方线程同步）

tc_iot_av_init

tc_iot_av_init

tc_iot_error_e tc_iot_av_init (const tc_iot_av_observer_s *observer)

初始化 AV 模块并注册观察者（内部拷贝 observer）
须在 login 成功且设备身份就绪后调用。

参数	描述
observer	事件观察者，不可为 NULL

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_INVALID_ARGUMENT	参数非法（空指针、非法取值等）
TC_IOT_ERR_ALREADY_INITIALIZED	重复初始化

TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪
TC_IOT_ERR_OUT_OF_MEMORY	内存不足
TC_IOT_ERR_POST_TASK_FAILED	投递到 iot 消息循环失败
TC_IOT_ERR_FAILURE	通用失败

tc_iot_av_deinit

tc_iot_av_deinit
tc_iot_error_e tc_iot_av_deinit (void)

反初始化 AV；同步清除单例后异步退订信令

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪

tc_iot_av_get_contacts

tc_iot_av_get_contacts
tc_iot_error_e tc_iot_av_get_contacts (const char *cursor, uint32_t limit, tc_iot_av_co
callback)

拉取联系人列表

参数	描述
cursor	分页游标；首页传 NULL 或空串
limit	单页条数上限

callback	结果回调；占位实现下不会触发

返回值	描述
TC_IOT_ERR_SUCCESS	占位实现固定返回成功

注意
当前为占位实现：同步返回 SUCCESS 且不触发 callback。

tc_iot_av_call

tc_iot_av_call
tc_iot_error_e tc_iot_av_call (const tc_iot_contact_s *contact, tc_iot_call_option_t op
tc_iot_av_operation_cb callback)

发起呼出
contact->user_id 必填；建议提供非 NULL callback。若已有非监控会话进行中，回调 TC_IOT_ERR_AV_ON_CALLING 。

参数	描述
contact	被叫联系人
option	媒体选项
callback	操作结果回调

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪
TC_IOT_ERR_INVALID_ARGUMENT	参数非法（空指针、非法取值等）
TC_IOT_ERR_OUT_OF_MEMORY	内存不足

TC_IOT_ERR_POST_TASK_FAILED	投递到 iot 消息循环失败

tc_iot_av_accept

tc_iot_error_e tc_iot_av_accept (const tc_iot_contact_s *contact, tc_iot_av_operation

接听呼入；仅在振铃态有效，否则回调 AV_ON_*

参数	描述
contact	主叫联系人
callback	操作结果回调，不可为 NULL

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪
TC_IOT_ERR_INVALID_ARGUMENT	参数非法（空指针、非法取值等）
TC_IOT_ERR_OUT_OF_MEMORY	内存不足
TC_IOT_ERR_POST_TASK_FAILED	投递到 iot 消息循环失败

tc_iot_av_reject

tc_iot_error_e tc_iot_av_reject (const tc_iot_contact_s *contact, tc_iot_av_operation

拒绝呼入；仅在振铃态有效

参数	描述
contact	主叫联系人
callback	操作结果回调，不可为 NULL

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪
TC_IOT_ERR_INVALID_ARGUMENT	参数非法（空指针、非法取值等）
TC_IOT_ERR_OUT_OF_MEMORY	内存不足
TC_IOT_ERR_POST_TASK_FAILED	投递到 iot 消息循环失败

tc_iot_av_hangup

tc_iot_av_hangup

tc_iot_error_e tc_iot_av_hangup (const tc_iot_contact_s *contact, tc_iot_av_operation

挂断当前会话；无对应会话时回调 TC_IOT_ERR_AV_ON_IDLE

参数	描述
contact	对端联系人
callback	操作结果回调，不可为 NULL

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪

TC_IOT_ERR_INVALID_ARGUMENT	参数非法（空指针、非法取值等）
TC_IOT_ERR_OUT_OF_MEMORY	内存不足
TC_IOT_ERR_POST_TASK_FAILED	投递到 iot 消息循环失败

tc_iot_create_video_channel

tc_iot_create_video_channel

tc_iot_av_channel_t *
tc_iot_create_video_channel (int channel_id, tc_iot_video_quality_e

创建视频推流通道

channel_id 全局唯一；同时最多 4 路。

参数	描述
channel_id	通道 ID
quality	画质档位

返回值	描述
tc_iot_av_channel_t *	通道句柄；失败返回 NULL（未 init / id 冲突 / 已满 / OOM）

tc_iot_create_audio_channel

tc_iot_create_audio_channel

tc_iot_av_channel_t *
tc_iot_create_audio_channel ()

创建全局唯一音频推流通道

已存在或未 init 时返回 NULL。多路监控须复用同一指针。

返回值	描述

tc_iot_av_channel_t *	通道句柄；失败返回 NULL

tc_iot_destroy_video_channel

```
tc_iot_error_e tc_iot_destroy_video_channel(tc_iot_av_channel_t *av_channel)
```

销毁视频通道

参数	描述
av_channel	视频通道

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_INVALID_ARGUMENT	参数非法（空指针、非法取值等）

tc_iot_destroy_audio_channel

```
tc_iot_error_e tc_iot_destroy_audio_channel(tc_iot_av_channel_t *av_channel)
```

销毁音频通道

参数	描述
av_channel	音频通道

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_INVALID_ARGUMENT	参数非法（空指针、非法取值等）

tc_iot_push_video_frame

tc_iot_error_e tc_iot_push_video_frame(tc_iot_av_channel_t *av_channel, const tc_iot_video_frame_t *frame)

推送一帧视频（调用方线程同步）

须为视频通道。格式见 tc_iot_def.h（Annex-B；IDR 含参数集）。

参数	描述
av_channel	视频通道
frame	视频帧

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_INVALID_ARGUMENT	参数非法（空指针、非法取值等）

tc_iot_push_audio_frame

tc_iot_error_e tc_iot_push_audio_frame(tc_iot_av_channel_t *av_channel, const tc_iot_audio_frame_t *frame)

推送一帧音频（调用方线程同步）

须为音频通道；仅接受 16-bit PCM。 同一通道生命周期内 sample_rate / channels 应保持稳定。

参数	描述
av_channel	音频通道
frame	音频帧

返回值	描述

TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_INVALID_ARGUMENT	参数非法（空指针、非法取值等）

ptz_command_e

ptz_command_e
云台控制命令

枚举	取值	描述
PTZ_CMD_UP	0	上仰
PTZ_CMD_DOWN	1	下俯
PTZ_CMD_LEFT	2	左转
PTZ_CMD_RIGHT	3	右转
PTZ_CMD_ZOOM_IN	4	放大
PTZ_CMD_ZOOM_OUT	5	缩小
PTZ_CMD_STOP	6	停止运动

tc_iot_av_channel_t

tc_iot_av_channel_t
不透明 AV 推流通道句柄

typedef struct tc_iot_av_channel_s tc_iot_av_channel_t

tc_iot_call_option_t

tc_iot_call_option_t
呼叫 / 监控媒体选项

typedef struct tc_iot_call_option_s tc_iot_call_option_t

tc_iot_av_operation_cb

tc_iot_av_operation_cb
呼叫操作完成回调（MQTT puback / 状态冲突结果）

typedef void(* tc_iot_av_operation_cb)(tc_iot_error_e error_code, const char *error_message)

参数	描述
error_code	结果码
error_message	错误描述，可为 NULL

tc_iot_av_contacts_cb

tc_iot_av_contacts_cb
联系人列表回调（当前 get_contacts 为占位，不会触发）

typedef void(* tc_iot_av_contacts_cb)(tc_iot_error_e error_code, const char *error_message, const tc_iot_contact_s *contacts, uint32_t count, const char *next_cursor)

宏定义

宏	取值	描述
MAX_CUSTOM_DATA_LENGTH	20	call_option.custom_data 最大字节数（含结尾 '\0' 由调用方自行保证）

AI 对话（tc_iot_aitalk.h）

AI 对话（AITalk）

线程模型

本头文件接口可在任意线程调用；多数操作投递到 iot 消息循环。 tc_iot_aitalk_deinit 为阻塞等待清理完成。

回调 / 通知模型

- observer：init 时保存；user_data 透传到各回调。
- on_receive_bot_audio 等媒体/文本回调可能来自通道线程，须尽快返回。
- on_error 尽量投递到 iot 消息循环。
- tc_iot_aitalk_operation_cb（start_speak / register_contacts）： 同步 SUCCESS 则一定回调；同步失败则一定不回调（非法 mode 例外：同步返回 INVALID_ARGUMENT 且同步调用 callback）。

生命周期

tc_iot_init → tc_iot_login → tc_iot_aitalk_init →（可选）register_contacts → start_speak → send_audio / send_text → stop_speak → tc_iot_aitalk_deinit

业务约束

- 当前仅支持 TC_IOT_AITALK_MODE_CONTINUOUS。
- init 的 codec 须为 AAC / OPUS / G722。
- send_audio：帧 codec 与 init 一致，或送 16-bit PCM（由 SDK 内部编码）。
- 推荐 16 kHz 单声道；frame_duration_ms 与 codec 匹配（如 20）。
- 下行 bot 音频格式由通道侧决定，以 on_receive_bot_audio 的 frame 字段为准（常见为 PCM）。

函数列表	描述
tc_iot_aitalk_init	初始化 AITalk
tc_iot_aitalk_deinit	反初始化；阻塞等待内部 teardown 完成
tc_iot_aitalk_start_speak	开始对话
tc_iot_aitalk_stop_speak	停止对话（异步投递，空闲时幂等成功）
tc_iot_aitalk_send_audio	发送一帧上行音频
tc_iot_aitalk_send_text	发送文本给 LLM

tc_iot_aitalk_register_contacts	注册可供机器人呼叫的联系人
tc_iot_aitalk_interrupt	打断机器人当前播报（连续模式下有效）

tc_iot_aitalk_init

tc_iot_aitalk_init
tc_iot_error_e tc_iot_aitalk_init (const tc_iot_aitalk_init_params_s *init_params, co
tc_iot_aitalk_observer_s *observer, void *user_data

初始化 AITalk

须已 tc_iot_init（存在 iot 消息循环）。

参数	描述
init_params	初始化参数
observer	事件观察者
user_data	透传到各 observer 回调

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_INVALID_ARGUMENT	参数非法（空指针、非法取值等）
TC_IOT_ERR_ALREADY_INITIALIZED	重复初始化
TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪
TC_IOT_ERR_OUT_OF_MEMORY	内存不足

tc_iot_aitalk_deinit

tc_iot_aitalk_deinit

tc_iot_error_e tc_iot_aitalk_deinit (void)

反初始化；阻塞等待内部 teardown 完成

若处于 Starting/Running，会先停止会话。

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪

tc_iot_aitalk_start_speak

tc_iot_aitalk_start_speak

tc_iot_error_e tc_iot_aitalk_start_speak (tc_iot_aitalk_mode_e mode, tc_iot_aitalk_ callback)

开始对话

仅支持 CONTINUOUS；会话已有效时立即成功回调。

参数	描述
mode	对话模式
callback	结果回调

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_INVALID_ARGUMENT	参数非法（空指针、非法取值等）
TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪
TC_IOT_ERR_OUT_OF_MEMORY	内存不足

TC_IOT_ERR_POST_TASK_FAILED	投递到 iot 消息循环失败
TC_IOT_ERR_FAILURE	通用失败

tc_iot_aitalk_stop_speak

tc_iot_error_e tc_iot_aitalk_stop_speak (void)

停止对话（异步投递，空闲时幂等成功）

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪
TC_IOT_ERR_OUT_OF_MEMORY	内存不足
TC_IOT_ERR_FAILURE	通用失败

tc_iot_aitalk_send_audio

tc_iot_error_e tc_iot_aitalk_send_audio (const tc_iot_audio_frame *frame)

发送一帧上行音频

同步 SUCCESS 表示任务已投递；未进房时内部可能丢弃。

参数	描述
frame	音频帧

返回值	描述
TC_IOT_ERR_SUCCESS	成功

TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪
TC_IOT_ERR_INVALID_ARGUMENT	参数非法（空指针、非法取值等）
TC_IOT_ERR_OUT_OF_MEMORY	内存不足
TC_IOT_ERR_FAILURE	通用失败

tc_iot_aitalk_send_text

tc_iot_aitalk_send_text

tc_iot_error_e tc_iot_aitalk_send_text (const char *text)

发送文本给 LLM

参数	描述
text	文本；长度上限见实现（当前 1024）

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪
TC_IOT_ERR_INVALID_ARGUMENT	参数非法（空指针、非法取值等）
TC_IOT_ERR_OUT_OF_MEMORY	内存不足
TC_IOT_ERR_FAILURE	通用失败

tc_iot_aitalk_register_contacts

tc_iot_aitalk_register_contacts

tc_iot_error_e tc_iot_aitalk_register_contacts (const tc_iot_contact_s *contacts, uint contact_count, tc_iot_aitalk_operation

注册可供机器人呼叫的联系人

count ∈ [1, TC_IOT_AITALK_MAX_CONTACT_COUNT]。结果缓存至下次会话；callback 报告缓存结果。

参数	描述
contacts	联系人数组
contact_count	数量
callback	结果回调

返回值	描述
TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪
TC_IOT_ERR_INVALID_ARGUMENT	参数非法（空指针、非法取值等）
TC_IOT_ERR_OUT_OF_MEMORY	内存不足
TC_IOT_ERR_FAILURE	通用失败

tc_iot_aitalk_interrupt

tc_iot_aitalk_interrupt

tc_iot_error_e tc_iot_aitalk_interrupt (void)

打断机器人当前播报（连续模式下有效）

返回值	描述

TC_IOT_ERR_SUCCESS	成功
TC_IOT_ERR_NOT_INITIALIZED	核心或业务模块尚未 init / 内部 impl 未就绪
TC_IOT_ERR_FAILURE	通用失败
TC_IOT_ERR_AITALK_*	• TC_IOT_ERR_AITALK_NOT_SUPPORT：AITalk：能力或模式不支持（亦可经 on_error）； • TC_IOT_ERR_AITALK_BOT_LEAVED：AITalk：机器人已离开会话； • TC_IOT_ERR_AITALK_INVALID_PARAM：AITalk：参数非法； • TC_IOT_ERR_AITALK_INTERNAL_ERROR：AITalk：内部错误； • TC_IOT_ERR_AITALK_NOT_ACTIVATED：AITalk：服务未开通 / 未激活； • TC_IOT_ERR_AITALK_SERVICE_EXPIRED：AITalk：服务已过期； • TC_IOT_ERR_AITALK_ASR_ERROR：AITalk：ASR 错误； • TC_IOT_ERR_AITALK_TTS_ERROR：AITalk：TTS 错误； • TC_IOT_ERR_AITALK_LLM_ERROR：AITalk：LLM 错误

tc_iot_aitalk_mode_e

tc_iot_aitalk_mode_e
对话模式

枚举	取值	描述
TC_IOT_AITALK_MODE_CONTINUOUS	0	连续对话（当前唯一可用模式）
TC_IOT_AITALK_MODE_PUSH_TO_TALK	1	按键说话；当前 start_speak 不接受该模式

tc_iot_aitalk_bot_state_e

tc_iot_aitalk_bot_state_e
机器人状态

枚举	取值	描述
TC_IOT_AITALK_BOT_STATE_LISTENING	1	正在聆听用户语音
TC_IOT_AITALK_BOT_STATE_THINKING	2	正在思考 / 推理
TC_IOT_AITALK_BOT_STATE_SPEAKING	3	正在播报回复
TC_IOT_AITALK_BOT_STATE_INTERRUPTED	4	播报被打断
TC_IOT_AITALK_BOT_STATE_FINISHED	5	本轮对话结束

tc_iot_aitalk_operation_cb

tc_iot_aitalk_operation_cb
异步操作结果回调（start_speak / register_contacts）

typedef void(* tc_iot_aitalk_operation_cb)(tc_iot_error_e error_code, const char *error_message)

宏定义

宏	取值	描述
TC_IOT_AITALK_BOT_ID_MAX_LEN	16	bot_id 最大长度（不含结尾 '\0'）
TC_IOT_AITALK_MAX_CONTACT_COUNT	10	register_contacts 单次最多联系人数

类型定义（tc_iot_def.h 及公共结构体）

音视频公共类型（AV / AITalk 共用）

帧所有权

tc_iot_*_frame 中的 data 由调用方持有。push / send 接口在返回前会拷贝 或同步消费完毕；回调中的 frame->data 仅在回调返回前有效，调用方如需 异步使用须自行拷贝。

格式约定

- AV 上行音频：仅 PCM，16-bit 小端交错（单声道连续样本，立体声 LRLR…）；data_size 须与 sample_rate/channels/frame_duration_ms 一致（推荐 20ms）。SDK 内部可再编码为 AAC 供 TRTC。
- AV 下行音频：observer 收到 PCM；sample_rate/channels 透传自解码帧（由远端编码决定；当前未按进房 obtain 参数重采样）。
- AV 上行视频：Annex-B 裸流（H264/H265）；不支持 B 帧。type=IDR 时须将 SPS/PPS（H265 另含 VPS）与 IDR 放在同一缓冲； 先送 IDR 再送 P 帧，否则对端可能无法起播。
- AITalk：init 指定编码格式；send_audio 可送同格式编码帧，或 PCM（init 为 AAC/OPUS/G722 时由 SDK 内部编码）。

tc_iot_audio_codec_e

tc_iot_audio_codec_e
音频编码类型

枚举	取值	描述
TC_IOT_AUDIO_CODEC_UNKNOWN	0	未知 / 未指定
TC_IOT_AUDIO_CODEC_PCM	1	PCM（AV push 仅接受此类型）
TC_IOT_AUDIO_CODEC_AAC	2	AAC
TC_IOT_AUDIO_CODEC_OPUS	3	Opus
TC_IOT_AUDIO_CODEC_G722	4	G.722

tc_iot_audio_sample_rate_e

tc_iot_audio_sample_rate_e
音频采样率（Hz）

枚举	取值	描述
TC_IOT_AUDIO_SAMPLE_RATE_UNKNOWN	0	未知 / 未指定
TC_IOT_AUDIO_SAMPLE_RATE_8000	8000	8 kHz
TC_IOT_AUDIO_SAMPLE_RATE_16000	16000	16 kHz（推荐）
TC_IOT_AUDIO_SAMPLE_RATE_32000	32000	32 kHz
TC_IOT_AUDIO_SAMPLE_RATE_48000	48000	48 kHz

tc_iot_audio_channel_e

tc_iot_audio_channel_e
音频声道数

枚举	取值	描述
TC_IOT_AUDIO_CHANNEL_UNKNOWN	0	未知 / 未指定
TC_IOT_AUDIO_CHANNEL_MONO	1	单声道
TC_IOT_AUDIO_CHANNEL_STEREO	2	立体声（样本交错 LRLR…）

tc_iot_video_codec_e

tc_iot_video_codec_e
视频编码类型

枚举	取值	描述
TC_IOT_VIDEO_CODEC_UNKNOWN	0	未知 / 未指定
TC_IOT_VIDEO_CODEC_H264	1	H.264 / AVC
TC_IOT_VIDEO_CODEC_H265	2	H.265 / HEVC
TC_IOT_VIDEO_CODEC_MJPEG	3	Motion JPEG

tc_iot_video_frame_type_e

tc_iot_video_frame_type_e
视频帧类型

枚举	取值	描述
TC_IOT_VIDEO_FRAME_TYPE_UNKNOWN	0	未知 / 未指定
TC_IOT_VIDEO_FRAME_TYPE_IDR	1	关键帧 / IDR（缓冲须含 SPS/PPS，H265 另含 VPS）
TC_IOT_VIDEO_FRAME_TYPE_P	2	P 帧（预测帧）

tc_iot_video_rotation_e

tc_iot_video_rotation_e
视频旋转角度（度）

枚举	取值	描述

TC_IOT_VIDEO_ROTATION_0	0	不旋转
TC_IOT_VIDEO_ROTATION_90	90	顺时针 90°
TC_IOT_VIDEO_ROTATION_180	180	180°
TC_IOT_VIDEO_ROTATION_270	270	顺时针 270°

tc_iot_video_quality_e

tc_iot_video_quality_e
视频画质档位（呼叫 option / 监控切换 / 建视频通道时使用）

枚举	取值	描述
TC_IOT_VIDEO_QUALITY_UNKNOWN	0	未知 / 未指定
TC_IOT_VIDEO_QUALITY_LD	1	流畅（LD）
TC_IOT_VIDEO_QUALITY_SD	2	标清（SD）
TC_IOT_VIDEO_QUALITY_HD	3	高清（HD）
TC_IOT_VIDEO_QUALITY_FHD	4	超清（FHD）

tc_iot_media_content_e

tc_iot_media_content_e
会话媒体内容类型

枚举	取值	描述
TC_IOT_MEDIA_CONTENT_UNKNOWN	0	未知 / 未指定
TC_IOT_MEDIA_CONTENT_AUDIO	1	仅音频

TC_IOT_MEDIA_CONTENT_VIDEO	2	仅视频
TC_IOT_MEDIA_CONTENT_AUDIO_VIDEO	3	音视频

结构体与联合体

data_model_data_item_t

data_model_data_item_t
物模型数据项

字段	类型	描述
data_id	char *	标识符（属性 / 事件参数 / 行为参数 ID）
data_type	data_model_data_type_t	数据类型
data_value	data_model_data_value_t	数据值；按 data_type 选用联合体字段

data_model_data_value_t

data_model_data_value_t
物模型数据值联合体（按 data_type 选用对应字段）

字段	类型	描述
value_bool	data_model_bool_t	DATA_MODEL_DATA_TYPE_BOOL
value_int	data_model_int_t	DATA_MODEL_DATA_TYPE_INT
value_float	data_model_float_t	DATA_MODEL_DATA_TYPE_FLOAT
value_string	data_model_string_t	DATA_MODEL_DATA_T

		YPE_STRING
value_enum	data_model_enum_t	DATA_MODEL_DATA_TYPE_ENUM
value_time	data_model_time_t	DATA_MODEL_DATA_TYPE_TIME

tc_iot_aitalk_audio_option_s

tc_iot_aitalk_audio_option_s
会话音频参数

字段	类型	描述
codec	tc_iot_audio_codec_e	会话编码；须为 AAC / OPUS / G722
sample_rate	tc_iot_audio_sample_rate_e	采样率；推荐 16000
frame_duration_ms	uint32_t	单帧时长（毫秒）；须与 codec 匹配，推荐 20

tc_iot_aitalk_init_params_s

tc_iot_aitalk_init_params_s
AITalk 初始化参数

字段	类型	描述
bot_id	char	机器人 ID；长度 ≤ TC_IOT_AITALK_BOT_ID_MAX_LEN
audio_option	tc_iot_aitalk_audio_option_s	会话音频参数
prompt_variables_json	const char *	可选；prompt 变量 JSON 字符串，生命周期须覆盖 init

		异步过程

tc_iot_aitalk_observer_s

tc_iot_aitalk_observer_s
AITalk 事件观察者

字段	类型	描述
on_receive_bot_audio	void(* on_receive_bot_audio)(const tc_iot_audio_frame *frame, void *user_data)	机器人下行音频；frame 仅在回调内有效
on_receive_bot_text	void(* on_receive_bot_text)(const char *text, void *user_data)	机器人下行文本
on_receive_asr_text	void(* on_receive_asr_text)(const char *text, void *user_data)	ASR 文本；仅最终结果（is_final）会回调
on_bot_state_changed	void(* on_bot_state_changed)(tc_iot_aitalk_bot_state_e old_state, tc_iot_aitalk_bot_state_e new_state, void *user_data)	机器人状态变化
on_launch_call	void(* on_launch_call)(const tc_iot_contact_s *contact, void *user_data)	机器人发起呼叫联系人（需配合 register_contacts）
on_error	void(* on_error)(tc_iot_error_e error_code, const char	会话或通道错误（含 AITALK_*）

	*error_msg, void *user_data)	

tc_iot_audio_frame

tc_iot_audio_frame
音频帧

字段	类型	描述
codec	tc_iot_audio_codec_e	音频编码类型
sample_rate	tc_iot_audio_sample_rate_e	采样率（Hz）
channels	tc_iot_audio_channel_e	声道数
frame_duration_ms	uint32_t	单帧时长（毫秒）；须与 data_size 匹配；推荐 20
data	uint8_t *	PCM 时为 16-bit 小端样本；调用方持有
data_size	size_t	data 字节数
pts_ms	uint64_t	呈现时间戳，毫秒；调用方保证单调

tc_iot_av_observer_s

tc_iot_av_observer_s
AV 事件观察者；字段可为 NULL 表示不关心该事件

字段	类型	描述
on_call_requested	void(* on_call_requested) (const tc_iot_contact_s *contact,	App 呼入邀请

	tc_iot_call_option_t option)	
on_call_accepted	void(* on_call_accepted) (const tc_iot_contact_s *contact)	对端接听（呼出场景）
on_call_rejected	void(* on_call_rejected) (const tc_iot_contact_s *contact)	对端拒绝
on_call_timeout	void(* on_call_timeout) (const tc_iot_contact_s *contact)	呼叫超时
on_call_hangup	void(* on_call_hangup) (const tc_iot_contact_s *contact)	对端挂断或会话中止
on_monitor_begin	void(* on_monitor_begin) (int channel_id, tc_iot_call_option_t option)	IPC 监控：App 开始拉流
on_monitor_switch	void(* on_monitor_switch) (int channel_id, tc_iot_video_quality_e quality)	IPC 监控：切换画质
on_monitor_end	void(* on_monitor_end) (int channel_id)	IPC 监控：结束拉流
on_ptz_command_received	void(* on_ptz_command_received) (int channel_id, ptz_command_e ptz_command, int speed)	收到云台控制命令；speed 为速度档位
on_audio_frame_received	void(* on_audio_frame_received) (const tc_iot_audio_frame *frame)	远端音频：PCM；sample_rate/channels 透传自解码帧（随远端编码，未按进房 obtain 重采样）；frame 仅回调内有效

on_video_frame_received	void(* on_video_frame_received) (const tc_iot_video_frame *frame)	远端视频；frame 仅回调内有效

tc_iot_call_option_s

tc_iot_call_option_s
呼叫 / 监控媒体选项

字段	类型	描述
media_content	tc_iot_media_content_e	媒体内容类型（音频 / 视频 / 音视频）
video_quality	tc_iot_video_quality_e	视频画质档位
custom_data	char	自定义附带数据；长度受 MAX_CUSTOM_DATA_LENGTH 约束

tc_iot_config_s

tc_iot_config_s
SDK 初始化配置

字段	类型	描述
storage_path	const char *	本地持久化目录；可为 NULL；非空时长度须合法（实现侧上限约 256）
log_level	tc_iot_log_level_e	日志输出阈值，见 tc_iot_log_level_e
on_log	void(* on_log) (tc_iot_log_level_e level, const char *log)	可选；为 NULL 时不向外抛日志文本

on_device_event	void(* on_device_event)(tc_iot_device_event_type_e event_type, const void *event_data)	可选；为 NULL 时不通知在线状态

tc_iot_contact_s

tc_iot_contact_s
联系人；呼叫相关 API 至少需要 user_id

字段	类型	描述
user_id	const char *	用户 ID，必填
user_name	const char *	显示名，可选
avatar_url	const char *	头像 URL，可选

tc_iot_data_model_action_t

tc_iot_data_model_action_t
行为（下行调用 + 上行输出）

字段	类型	描述
action_id	char *	行为 ID
token	char *	云端下发的请求 token，回复时须原样带回
timestamp	uint32_t	时间戳
action_input_data_num	int	输入参数个数
action_input_data_list	data_model_data_item_t *	输入参数列表
action_output_data_num	int	输出参数个数；在 on_receive_new_action

		_cb 内填写
action_output_data_list	data_model_data_item_t *	输出参数列表；在 on_receive_new_action_cb 内填写后由 SDK 回复云端

tc_iot_data_model_callback

tc_iot_data_model_callback
物模型回调表

字段	类型	描述
on_report_property_result_cb	void(* on_report_property_result_cb) (char *property_id, void *user_data, data_model_report_result_t result_code)	属性上报结果（含超时）；user_data 为 report_property 传入值
on_receive_property_changed_cb	void(* on_receive_property_changed_cb) (char *property_id, tc_iot_data_model_property_t *data_model_property)	云端下发属性变更
on_report_event_result_cb	void(* on_report_event_result_cb) (char *event_id, void *user_data, data_model_report_result_t result_code)	事件上报结果（含超时）；user_data 为 report_event 传入值
on_receive_new_action_cb	int(* on_receive_new_action_cb) (char *action_id, tc_iot_data_model_acti	收到行为调用。

	on_t *data_model_action)	

tc_iot_data_model_event_t

tc_iot_data_model_event_t
事件

字段	类型	描述
event_id	char *	事件 ID
event_type	data_model_event_type_t	事件类型
event_data_num	int	event_data_list 元素个数
event_data_list	data_model_data_item_t *	事件参数列表

tc_iot_data_model_property_t

tc_iot_data_model_property_t
属性

字段	类型	描述
property	data_model_data_item_t	属性数据项

tc_iot_device_info_s

tc_iot_device_info_s
设备三元组（login 用）

字段	类型	描述
product_id	const char *	产品 ID，必填
device_id	const char *	设备名 / 设备 ID，必填
device_secret	const char *	设备密钥，必填
region	const char *	地域字符串（如 "ap-guangzhou"）；校验存在，当前登录链路未使用

tc_iot_dynamic_register_param_s

tc_iot_dynamic_register_param_s
动态注册参数

字段	类型	描述
product_id	const char *	产品 ID，必填
device_id	const char *	设备名 / 设备 ID，必填
product_secret	const char *	产品密钥；长度须满足 [16, 64)

tc_iot_video_frame

tc_iot_video_frame
视频帧（Annex-B 裸流）

字段	类型	描述
codec	tc_iot_video_codec_e	视频编码类型
type	tc_iot_video_frame_type_e	IDR 缓冲须含 SPS/PPS（H265 含 VPS）；P 帧不含参数集即可

rotation	tc_iot_video_rotation_e	画面旋转角度
width	uint32_t	像素宽度
height	uint32_t	像素高度
data	uint8_t *	Annex-B 裸流；调用方持有；不可为 NULL，data_size > 0
data_size	size_t	data 字节数
pts_ms	uint64_t	呈现时间戳，毫秒；调用方保证单调（热路径不做校验）

错误码（tc_iot_err.h）

SDK 统一错误码

0 为成功，负数为失败。各模块返回值子集以对应头文件函数注释为准； 此处给出全量枚举含义。

tc_iot_error_e

tc_iot_error_e
SDK 统一错误码

枚举	取值	描述
TC_IOT_ERR_SUCCESS	0	成功
TC_IOT_ERR_FAILURE	-1	通用失败
TC_IOT_ERR_OUT_OF_MEMORY	-2	内存不足
TC_IOT_ERR_INVALID_ARGUMENT	-3	参数非法（空指针、非法取值等）
TC_IOT_ERR_NOT_INITIALIZED	-4	核心或业务模块尚未 init / 内部 impl 未就绪
TC_IOT_ERR_ALREADY_INITIALIZED	-5	重复初始化
TC_IOT_ERR_NOT_CONNECTED	-6	MQTT 未连接，或操作因登出被取消

TC_IOT_ERR_ALREADY_CONNECTED	-7	已处于已连接状态（如重复 login）
TC_IOT_ERR_CONNECT_FAILED	-8	MQTT / 网络连接失败（非 CONNACK 细分码）
TC_IOT_ERR_POST_TASK_FAILED	-9	投递到 iot 消息循环失败
TC_IOT_ERR_AV_ON_IDLE	-10	AV：空闲，无对应会话（操作与状态机冲突）
TC_IOT_ERR_AV_ON_OUTGOING	-11	AV：正在呼出
TC_IOT_ERR_AV_ON_RINGING	-12	AV：正在振铃（呼入）
TC_IOT_ERR_AV_ON_CALLING	-13	AV：已在通话中
TC_IOT_ERR_AV_ON_IPC_MONITOR	-14	AV：处于 IPC 监控会话
TC_IOT_ERR_DYNAMIC_REGISTER_BUSY	-20	动态注册：已有注册进行中（含阻塞 deinit）
TC_IOT_ERR_DYNAMIC_REGISTER_REQUEST_FAILED	-21	动态注册：请求发送或网络失败
TC_IOT_ERR_DYNAMIC_REGISTER_RESPONSE_INVALID	-22	动态注册：响应解析失败或内容非法
TC_IOT_ERR_NTP_TIME_NOT_READY	-30	NTP 尚未同步完成（ONLINE 之后仍可能短暂返回此值）
TC_IOT_ERR_AITALK_NOT_SUPPORT	-200	AITalk：能力或模式不支持（亦可经 on_error）
TC_IOT_ERR_AITALK_BOT_LEAVED	-201	AITalk：机器人已离开会话
TC_IOT_ERR_AITALK_INVALID_PARAM	-202	AITalk：参数非法
TC_IOT_ERR_AITALK_INTERNAL_ERROR	-203	AITalk：内部错误
TC_IOT_ERR_AITALK_NOT_ACTIVATED	-204	AITalk：服务未开通 / 未激活

TC_IOT_ERR_AITALK_SERVICE_EXPIRED	-205	AITalk：服务已过期
TC_IOT_ERR_AITALK_ASR_ERROR	-206	AITalk：ASR 错误
TC_IOT_ERR_AITALK_TTS_ERROR	-207	AITalk：TTS 错误
TC_IOT_ERR_AITALK_LLM_ERROR	-208	AITalk：LLM 错误
TC_IOT_ERR_CONNECT_UNACCEPTABLE_PROTOCOL_VERSION	-301	MQTT CONNACK：不接受的协议版本
TC_IOT_ERR_CONNECT_IDENTIFIER_REJECTED	-302	MQTT CONNACK：客户端标识被拒绝
TC_IOT_ERR_CONNECT_SERVER_UNAVAILABLE	-303	MQTT CONNACK：服务端不可用
TC_IOT_ERR_CONNECT_BAD_USERNAME_OR_PASSWORD	-304	MQTT CONNACK：用户名或密码错误
TC_IOT_ERR_CONNECT_NOT_AUTHORIZED	-305	MQTT CONNACK：未授权

Android TXIoTEngine

最近更新时间：2026-07-30 12:00:38

概述

`TXIoTEngine` 是整个 SDK 的入口，通过 `getInstance(Context)` 获取单例，使用前须先调用 `login` 完成身份验证。

> **说明：**
> 线程模型：SDK 接口可在任意线程调用；回调均在主线程（UI 线程）触发，调用方可在回调中直接操作 UI。

SDK 模块总览：

模块	说明
TXIoTEngine 引擎管理	登录、获取各个业务对象。
TXIoTFamilyManager 家庭与房间管理	家庭、房间、成员管理。
TXIoTDeviceManager 设备管理	设备绑定/解绑、分享、指令下发、属性查询。
TXIoTCallSession 音视频通话	音视频通话。
TXIoTMonitorSession 监控直播	监控直播（拉流/对讲/云台/截图/录像）。
公共类型与枚举	错误码、通用枚举、公共数据结构、回调接口。

方法列表

方法签名	描述
getInstance	获取 TXIoTEngine 单例。
addListener	添加引擎事件监听器。
removeListener	移除引擎事件监听器。
login	登录 SDK（异步）。
logout	登出 SDK。

getLoginUserInfo	获取当前登录用户信息。
getFamilyManager	获取家庭管理器，未登录时返回 null。
getDeviceManager	获取设备管理器，未登录时返回 null。
getCallSession	获取音视频通话会话，未登录时返回 null。
getMonitorSession	获取实时监控会话，未登录时返回 null。

getInstance

getInstance

TXIoTEngine getInstance (Context context)

获取 TXIoTEngine 单例

参数	描述
context	`Context`；Android 上下文。

返回值	描述
TXIoTEngine	返回 TXIoTEngine 单例实例。

addListener

addListener

void addListener (TXIoTEngineListener listener)

添加引擎事件监听器

参数	描述
listener	`TXIoTEngineListener`；监听器实例。

removeListener

removeListener

void removeListener (TXIoTEngineListener listener)

移除引擎事件监听器

参数	描述
listener	`TXIoTEngineListener`；监听器实例。

login

login

void login (String appKey, String userId, TXIoTUserSignature user

登录 SDK（异步）

参数	描述
appKey	`String`；应用 Key。
userId	`String`；用户 ID。
userSignature	`TXIoTUserSignature`；用户身份签名（含 requestId / timestamp / nonce / signature）。

logout

logout

void logout ()

登出 SDK

getLoginUserInfo

getLoginUserInfo

TXIoTUserInfo getLoginUserInfo ()

获取当前登录用户信息

返回值	描述
TXIoTUserInfo	返回 TXIoTUserInfo 对象；未登录时返回 `null`。

getFamilyManager

getFamilyManager

TXIoTFamilyManager getFamilyManager ()

获取家庭管理器（登录后可用）

返回值	描述
TXIoTFamilyManager	返回 TXIoTFamilyManager 实例；未登录时返回 `null`。

getDeviceManager

getDeviceManager

TXIoTDeviceManager getDeviceManager ()

获取设备管理器（登录后可用）

返回值	描述
TXIoTDeviceManager	返回 TXIoTDeviceManager 实例；未登录时返回 `null`。

getCallSession

getCallSession
TXIoTCallSession
getCallSession
()

获取音视频通话会话（登录后可用）

返回值	描述
TXIoTCallSession	返回 TXIoTCallSession 实例；未登录时返回 `null`。

getMonitorSession

getMonitorSession
TXIoTMonitorSession
getMonitorSession
()

获取实时监控会话（登录后可用）

返回值	描述
TXIoTMonitorSession	返回 TXIoTMonitorSession 实例；未登录时返回 `null`。

引擎事件回调

`TXIoTEngineListener` 是抽象类，所有回调方法均有默认空实现，调用方可按需覆写。

TXIoTEngineListener

回调方法	说明
`onLoginSuccess()`	登录成功。登录成功后各业务模块 getter 生效。

`onLoginFailure(TXIoTErrorCode errCode, String errMsg)`	登录失败。errCode 为错误码，errMsg 为错误描述。
`onLogout()`	登出完成。各业务模块 getter 失效。
`onUserSignatureExpired()`	用户签名已过期，需重新获取签名后调用 `login`。
`onReceivePushMessage(TXIoTPushMessage pushMessage)`	收到服务器推送消息，包含设备上下线等状态变更通知。

引擎相关数据结构与枚举

TXIoTPushMessage

TXIoTPushMessage
设备推送消息，包含消息时间、类型和关联设备

字段	说明
messageTime	消息时间戳。
type	消息类型。
subType	消息子类型。
deviceId	关联的设备 ID。

TXIoTPushMessageType

TXIoTPushMessageType

枚举	说明
STATUS_CHANGE	设备状态变更。

TXIoTPushMessageSubType

TXIoTPushMessageSubType

枚举	说明
ONLINE	设备上线。
OFFLINE	设备离线。

TXIoTUserSignature

TXIoTUserSignature

字段	说明
requestId	请求 ID，由服务端签发。
timestamp	签名的 Unix 秒级时间戳。
nonce	签名随机数。
signature	签名字符串，由服务端签发。

TXIoTEngineDef

最近更新时间：2026-07-30 12:00:38

概述

本文档描述 IoT App SDK 在 Android 端的公共类型与枚举定义，包含错误码、通用枚举、公共数据结构与通用回调等，定义于 `TXIoTEngineDef` 接口中。

> **说明：**
> `TXIoTEngineDef` 是各模块共用的基础类型定义接口，其他模块均通过 SDK 的公共依赖引入。各模块文档见：
> - TXIoTEngine：引擎管理。
> - TXIoTFamilyManager：家庭与房间管理。
> - TXIoTDeviceManager：设备管理。
> - TXIoTCallSession：音视频通话。
> - TXIoTMonitorSession：实时监控。

错误码 — TXIoTErrorCode

TXIoTErrorCode

错误码	说明
ERR_SUCCESS	成功。
ERR_FAILED	通用失败。
ERR_INVALID_PARAMETER	参数非法。
ERR_INVALID_ACCESS_TOKEN	Access Token 无效。
ERR_RATE_LIMITED	请求频率超限。
ERR_UNAUTHORIZED_OPERATION	未授权操作。
ERR_DEVICE_BOUND	设备已绑定。
ERR_BIND_TOKEN_NOT_EXIST	绑定令牌不存在。
ERR_BIND_TOKEN_IS_EXPIRED	绑定令牌已过期。

ERR_BIND_TOKEN_NO_PAIR_WITH_DEVICE	绑定令牌与设备不匹配。
ERR_INVALID_MESSAGE_ID	消息 ID 无效。
ERR_FAMILY_DEVICE_LIMIT_EXCEEDED	家庭设备数量超限。
ERR_CAN_NOT_BIND_SAME_FAMILY	不能绑定到同一家庭。
ERR_FAMILY_NOT_EXIST	家庭不存在。
ERR_PRODUCT_NOT_EXIST	产品不存在。
ERR_DEVICE_NOT_EXIST	设备不存在。
ERR_DEVICE_OFFLINE	设备离线。
ERR_CAMERA_START_FAIL	摄像头启动失败。
ERR_CAMERA_NOT_AUTHORIZED	摄像头权限未授权。
ERR_CAMERA_OCCUPY	摄像头被占用。
ERR_MIC_START_FAIL	麦克风启动失败。
ERR_MIC_NOT_AUTHORIZED	麦克风权限未授权。
ERR_MIC_OCCUPY	麦克风被占用。
ERR_SPEAKER_START_FAIL	扬声器启动失败。
ERR_RECORD_DURATION_TOO_SHORT	录制时长过短。
ERR_FILE_PATH_UNWRITABLE	文件路径不可写。
ERR_FILE_PATH_ALREADY_EXIST	文件路径已存在。
ERR_DEVICE_SWITCH_TO_VOIP	设备切换到 VOIP 模式。

通用枚举

TXIoTFamilyRole

TXIoTFamilyRole
家庭角色枚举。在 TXIoTFamilyManager 的成员管理接口中使用。

枚举	说明
UNKNOWN	未知。
MEMBER	普通成员。
ADMIN	管理员。

公共数据结构

以下类型定义于 `TXIoTEngineDef` 接口中，作为 SDK 内部共用数据类型。

TXIoTDeviceId

TXIoTDeviceId
设备 ID，用于唯一标识一个设备。在 TXIoTDeviceManager 的绑定、查询接口中使用。

字段	说明
productId	产品 ID。
deviceName	设备名称。

TXIoTDeviceStatus

TXIoTDeviceStatus
设备在线状态。包含当前在线情况、累计在线时长和上线时间戳。在 TXIoTDeviceManager 的设备信息中包含此字段。

字段	说明
isOnline	是否在线。
onlineDuration	累计在线时长（单位：秒）。

bringOnlineTime	上线时间戳（Unix 秒级时间戳）。

TXIoTDeviceInfo

TXIoTDeviceInfo
设备信息。在 TXIoTDeviceManager 的绑定、设备列表、被分享设备列表等查询接口中返回。

字段	说明
deviceId	设备 ID。
aliasName	设备别名。
familyId	所属家庭 ID。
roomId	所属房间 ID。
iconUrl	设备图标 URL。
createTime	创建时间（Unix 秒级时间戳）。
updateTime	更新时间（Unix 秒级时间戳）。
status	设备在线状态。

TXIoTFamilyInfo

TXIoTFamilyInfo
家庭信息。在 TXIoTFamilyManager 的创建、查询接口中使用。

字段	说明
familyId	家庭 ID。
name	家庭名称。
createTime	创建时间（Unix 秒级时间戳）。
updateTime	更新时间（Unix 秒级时间戳）。

role	当前用户在该家庭中的角色。

TXIoTRoomInfo

TXIoTRoomInfo
房间信息。房间是家庭下的二级分组，在 TXIoTFamilyManager 的房间管理接口中使用。

字段	说明
roomId	房间 ID。
name	房间名称。
familyId	所属家庭 ID。
deviceCount	房间内设备数量。
createTime	创建时间（Unix 秒级时间戳）。
updateTime	更新时间（Unix 秒级时间戳）。

TXIoTUserInfo

TXIoTUserInfo
用户信息。在 TXIoTFamilyManager 的成员管理和 TXIoTDeviceManager 的设备分享接口中使用。

字段	说明
userId	用户 ID。
nickName	用户昵称。
avatarUrl	用户头像 URL。
role	用户角色。
deviceBindTime	设备绑定时间（Unix 秒级时间戳）。

TXIoTPageResult

TXIoTPageResult
分页查询结果。`T` 为数据列表元素类型。在 TXIoTDeviceManager 的设备列表分页查询中使用：首次查询时 `nextPageToken` 传空，结果中返回的 `nextPageToken` 用于获取下一页。

字段	说明
nextPageToken	下一页游标；首次查询传空字符串，取结果中返回值用于获取下一页；为空字符串表示已到最后一页。
dataList	当前页数据列表。

通用回调

TXIoTCallback

TXIoTCallback
有返回值的异步回调。在所有业务模块（TXIoTDeviceManager、TXIoTFamilyManager 等）中广泛使用。`T` 为回调成功时返回的结果类型：`onSuccess(T result)` 携带结果数据；`onError(TXIoTErrorCode, String)` 携带错误码与描述信息。

```
interface TXIoTCallback<T> {
    void onSuccess(T result);
    void onError(TXIoTErrorCode errorCode, String errorMessage);
}
```

TXIoTFamilyManager

最近更新时间：2026-07-30 12:00:38

概述

TXIoTFamilyManager 负责家庭的创建/删除/更新、房间的创建/删除/重命名，以及家庭成员的邀请与管理。
TXIoTFamilyManager 通过 TXIoTEngine.getFamilyManager 获取实例，需在登录成功后调用。

> **说明：**
> 家庭是设备的顶层组织单元，房间是家庭下的二级分组。用户创建家庭后自动成为 ADMIN，可通过 `createFamilyInviteToken` 生成邀请令牌，其他用户通过 `joinFamilyAsMember` 加入。

方法列表

方法签名	描述
createFamily	创建家庭。
deleteFamily	删除家庭。
getFamilyList	获取当前用户的家庭列表。
updateFamilyInfo	更新家庭信息（如名称）。
createRoom	在指定家庭下创建房间。
deleteRoom	删除指定房间。
setRoomName	设置/修改房间名称。
getRoomList	获取指定家庭的房间列表。
createFamilyInviteToken	创建家庭邀请令牌。
joinFamilyAsMember	通过令牌加入家庭。
removeMemberFromFamily	从家庭中移除成员。
getMemberList	获取家庭成员列表。

createFamily

createFamily

void createFamily (String name, TXIoTCallback<TXIoTFamilyInfo> callbac

创建家庭

参数	描述
name	`String`；名称。
callback	TXIoTCallback<TXIoTFamilyInfo>；异步回调，通过 onSuccess / onError 回调通知结果。

deleteFamily

deleteFamily

void deleteFamily (String familyId, TXIoTCallback<Void> callback)

删除家庭

参数	描述
familyId	`String`；家庭 ID。
callback	TXIoTCallback<Void>；异步回调，通过 onSuccess / onError 通知结果。

getFamilyList

getFamilyList

void getFamilyList (TXIoTCallback<List<TXIoTFamilyInfo>> callback)

获取当前用户的家庭列表

参数	描述

callback	TXIoTCallback<List<TXIoTFamilyInfo>>；异步回调，通过 onSuccess / onError 通知结果。

updateFamilyInfo

```
void updateFamilyInfo (TXIoTFamilyInfo newFamilyInfo, TXIoTCallback<Void>
```

更新家庭信息（如名称）

参数	描述
newFamilyInfo	TXIoTFamilyInfo； TXIoTFamilyInfo 类型对象。
callback	TXIoTCallback<Void>；异步回调，通过 onSuccess / onError 通知结果。

createRoom

```
void createRoom (String familyId, String name, TXIoTCallback<TXIoTRoo
```

在指定家庭下创建房间

参数	描述
familyId	`String`；家庭 ID。
name	`String`；家庭名称。
callback	TXIoTCallback<TXIoTRoomInfo>；异步回调，通过 onSuccess / onError 通知结果。

deleteRoom

deleteRoom
void deleteRoom (String familyId, String roomId, TXIoTCallback<Void> ca

删除指定房间

参数	描述
familyId	`String`；家庭 ID。
roomId	`String`；房间 ID。
callback	TXIoTCallback<Void>；异步回调，通过 onSuccess / onError 通知结果。

setRoomName

setRoomName
void setRoomName (String familyId, String roomId, String name, TXIoTCallb
callback)

设置/修改房间名称

参数	描述
familyId	`String`；家庭 ID。
roomId	`String`；房间 ID。
name	`String`；房间名称。
callback	TXIoTCallback<Void>；异步回调，通过 onSuccess / onError 通知结果。

getRoomList

getRoomList

```
void getRoomList (String familyId, TXIoTCallback<List<TXIoTRoomInfo>>
```

获取指定家庭的房间列表

参数	描述
familyId	`String`；家庭 ID。
callback	TXIoTCallback<List<TXIoTRoomInfo>>；异步回调，通过 onSuccess / onError 通知结果。

createFamilyInviteToken

createFamilyInviteToken

```
void createFamilyInviteToken (String familyId, TXIoTCallback<String> callback)
```

创建家庭邀请令牌

参数	描述
familyId	`String`；家庭 ID。
callback	TXIoTCallback<String>；异步回调，通过 onSuccess / onError 通知结果。

joinFamilyAsMember

joinFamilyAsMember

```
void joinFamilyAsMember (String inviteToken, TXIoTCallback<Void> callback)
```

通过令牌加入家庭

参数	描述

inviteToken	`String`；邀请令牌。
callback	TXIoTCallback<Void>；异步回调，通过 onSuccess / onError 通知结果。

removeMemberFromFamily

```
void removeMemberFromFamily (String familyId, String userId, TXIoTCallback<Void> ca
```

从家庭中移除成员

参数	描述
familyId	`String`；家庭 ID。
userId	`String`；用户 ID。
callback	TXIoTCallback<Void>；异步回调，通过 onSuccess / onError 通知结果。

getMemberList

```
void getMemberList (String familyId, TXIoTCallback<List<TXIoTUserInfo>>
```

获取家庭成员列表

参数	描述
familyId	`String`；家庭 ID。
callback	TXIoTCallback<List<TXIoTUserInfo>>；异步回调，通过 onSuccess / onError 通知结果。

TXIoTDeviceManager

最近更新时间：2026-07-30 12:00:38

概述

TXIoTDeviceManager 负责设备的绑定/解绑、房间归属、设备分享、指令下发、属性查询与别名修改等。
TXIoTDeviceManager 通过 TXIoTEngine.getDeviceManager 获取实例，需在登录成功后调用。

> **说明：**
> 设备必须先绑定到家庭后才能进行后续操作。绑定方式有两种：
> - (1) 通过 `bindDevice` 使用设备绑定签名直接绑定。
> - (2) 通过 `bindDeviceSharedWithMe` 接受他人分享的设备。

方法列表

方法签名	描述
bindDevice	绑定设备到家庭。
unbindDevice	从家庭解绑设备。
addDeviceToRoom	将设备添加到房间。
removeDeviceFromRoom	将设备从房间移除。
getDeviceList	获取家庭设备列表（支持分页）。
createDeviceSharingToken	创建设备分享令牌。
bindDeviceSharedWithMe	绑定别人分享给我的设备。
unbindDeviceSharedWithMe	取消绑定别人分享给我的设备。
getDeviceSharedUsers	获取设备已分享的用户列表。
removeDeviceSharedUser	移除设备已分享用户。
getDeviceListSharedWithMe	获取别人分享给我的设备列表（支持分页）。
sendCommand	向设备发送指令。
getProperties	查询设备属性。
modifyAliasName	修改设备别名。

bindDevice

bindDevice
void bindDevice (String familyId, String deviceBindSignature, TXIoTCallback<TXIoTDeviceInfo> callback)

绑定设备到家庭

参数	描述
familyId	`String`；家庭 ID。
deviceBindSignature	`String`；设备绑定签名。
callback	TXIoTCallback<TXIoTDeviceInfo>；异步回调，通过 onSuccess / onError 通知结果。

unbindDevice

unbindDevice
void unbindDevice (String familyId, TXIoTDeviceId deviceId, TXIoTCallbacl

从家庭解绑设备

参数	描述
familyId	`String`；家庭 ID。
deviceId	TXIoTDeviceId；设备标识（含 productId / deviceName）。
callback	TXIoTCallback<Void>；异步回调，通过 onSuccess / onError 通知结果。

addDeviceToRoom

addDeviceToRoom
void addDeviceToRoom (TXIoTDeviceId deviceId, String familyId, String roomId, TXIoTCallback<Void> callback)

将设备添加到房间

参数	描述
deviceId	TXIoTDeviceId；设备标识（含 productId / deviceName）。
familyId	`String`；家庭 ID。
roomId	`String`；房间 ID。
callback	TXIoTCallback<Void>；异步回调，通过 onSuccess / onError 通知结果。

removeDeviceFromRoom

removeDeviceFromRoom
void removeDeviceFromRoom (TXIoTDeviceId deviceId, String familyId, TXIoTCallbacl

将设备从房间移除

参数	描述
deviceId	TXIoTDeviceId；设备标识（含 productId / deviceName）。
familyId	`String`；家庭 ID。
callback	TXIoTCallback<Void>；异步回调，通过 onSuccess / onError 通知结果。

getDeviceList

getDeviceList
void getDeviceList

(String familyId, String nextPageToken, TXIoTCallback<TXIoTPageResult<TXIoTDeviceInfo>>

获取家庭设备列表（支持分页）

参数	描述
familyId	`String`；家庭 ID。
nextPageToken	`String`；分页令牌，首次传空字符串。
callback	TXIoTCallback<TXIoTPageResult<TXIoTDeviceInfo>>；异步回调，通过 onSuccess / onError 通知结果。

createDeviceSharingToken

createDeviceSharingToken
void createDeviceSharingToken

(String family_id, TXIoTDeviceId deviceId, TXIoTCallba callback)

创建设备分享令牌

参数	描述
family_id	`String`；家庭 ID。
deviceId	TXIoTDeviceId；设备标识（含 productId / deviceName）。
callback	TXIoTCallback<String>；异步回调，通过 onSuccess / onError 通知结果。

bindDeviceSharedWithMe

bindDeviceSharedWithMe
void bindDeviceSharedWithMe (TXIoTDeviceId deviceId, String shareToken, TXIoTCall callback)

绑定别人分享给我的设备

参数	描述
deviceId	TXIoTDeviceId；设备标识（含 productId / deviceName）。
shareToken	`String`；分享令牌。
callback	TXIoTCallback<Void>；异步回调，通过 onSuccess / onError 通知结果。

unbindDeviceSharedWithMe

unbindDeviceSharedWithMe
void unbindDeviceSharedWithMe (TXIoTDeviceId deviceId, TXIoTCallback<Void> callbac

取消绑定别人分享给我的设备

参数	描述
deviceId	TXIoTDeviceId；设备标识（含 productId / deviceName）。
callback	TXIoTCallback<Void>；异步回调，通过 onSuccess / onError 通知结果。

getDeviceSharedUsers

getDeviceSharedUsers

void getDeviceSharedUsers	(TXIoTDeviceId deviceId, TXIoTCallback<List<TXIoTUs callback)

获取设备已分享的用户列表

参数	描述
deviceId	TXIoTDeviceId；设备标识（含 productId / deviceName）。
callback	TXIoTCallback<List<TXIoTUserInfo>>；异步回调，通过 onSuccess / onError 通知结果。

removeDeviceSharedUser

void removeDeviceSharedUser	(TXIoTDeviceId deviceId, String userId, TXIoTCallback<

移除设备已分享用户

参数	描述
deviceId	TXIoTDeviceId；设备标识（含 productId / deviceName）。
userId	`String`；用户 ID。
callback	TXIoTCallback<Void>；异步回调，通过 onSuccess / onError 通知结果。

getDeviceListSharedWithMe

void getDeviceListSharedWithMe	(String nextPageToken, TXIoTCallback<TXIoTPageResult<TXIoTDeviceInfo>>

获取别人分享给我的设备列表（支持分页）

参数	描述
nextPageToken	`String`；分页令牌，首次传空字符串。
callback	TXIoTCallback<TXIoTPageResult<TXIoTDeviceInfo>>；异步回调，通过 onSuccess / onError 通知结果。

sendCommand

sendCommand
void sendCommand (TXIoTDeviceId deviceId, String jsonData, TXIoTCallbac callback)

向设备发送指令

参数	描述
deviceId	TXIoTDeviceId；设备标识（含 productId / deviceName）。
jsonData	`String`；JSON 格式的指令数据。
callback	TXIoTCallback<String>；异步回调，通过 onSuccess / onError 通知结果。

getProperties

getProperties
void getProperties (TXIoTDeviceId deviceId, TXIoTCallback<String> callba

查询设备属性

参数	描述
deviceId	TXIoTDeviceId；设备标识（含 productId / deviceName）。

callback	TXIoTCallback<String>；异步回调，通过 onSuccess / onError 通知结果。

modifyAliasName

modifyAliasName
void modifyAliasName (TXIoTDeviceId deviceId, String aliasName, TXIoTCallb callback)

修改设备别名

参数	描述
deviceId	TXIoTDeviceId；设备标识（含 productId / deviceName）。
aliasName	`String`；设备别名。
callback	TXIoTCallback<Void>；异步回调，通过 onSuccess / onError 通知结果。

TXIoTCallSession

最近更新时间：2026-07-30 12:00:38

概述

TXIoTCallSession 负责与设备之间的音视频通话，包括发起呼叫、挂断、远端画面渲染、麦克风/摄像头控制及音频输出设备选择。TXIoTCallSession 通过 TXIoTEngine.getCallSession 获取实例，需在登录成功后调用。

> **说明：**
> 回调模型：通话事件通过 `TXIoTCallListener` 接口回调，所有 Listener 回调在主线程触发，公共类型与错误码定义见 TXIoTEngineDef 。

方法列表

方法签名	描述
addListener	添加通话事件监听器。
removeListener	移除通话事件监听器。
callDevice	向设备发起通话（音频/视频）。
hangup	挂断当前通话。
startRemoteView	开始渲染远端视频。
stopRemoteView	停止渲染远端视频。
openMicrophone	打开麦克风。
closeMicrophone	关闭麦克风。
openCamera	打开摄像头并设置本地预览视图。
closeCamera	关闭摄像头。
switchCamera	切换摄像头（前后置）。
selectAudioPlaybackDevice	选择音频播放设备（扬声器/听筒）。

addListener

addListener

void addListener (TXIoTCallListener listener)

添加通话事件监听器

参数	描述
listener	`TXIoTCallListener`；监听器实例。

removeListener

removeListener

void removeListener (TXIoTCallListener listener)

移除通话事件监听器

参数	描述
listener	`TXIoTCallListener`；监听器实例。

callDevice

callDevice

void callDevice (TXIoTDeviceId deviceId, TXIoTCallMediaType callType

向设备发起通话（音频/视频）

参数	描述
deviceId	TXIoTDeviceId；设备标识（含 productId / deviceName）。

callType	TXIoTCallMediaType；通话类型（音频/视频）。

hangup

```
void hangup()
```

挂断当前进行中的通话

startRemoteView

```
void startRemoteView(TXIoTCallUser callUser, TXCloudVideoView videoView)
```

开始渲染远端视频

参数	描述
callUser	TXIoTCallUser；参与通话的用户。
videoView	`TXCloudVideoView`；视频渲染视图。

stopRemoteView

```
void stopRemoteView(TXIoTCallUser callUser)
```

停止渲染远端视频

参数	描述

callUser	TXIoTCallUser；参与通话的用户。

openMicrophone

```
void openMicrophone()
```

打开麦克风

closeMicrophone

```
void closeMicrophone()
```

关闭麦克风

openCamera

```
void openCamera(TXIoTCamera camera, TXCloudVideoView videoView)
```

打开摄像头并设置本地预览视图

参数	描述
camera	TXIoTCamera；摄像头（前置/后置）。
videoView	`TXCloudVideoView`；视频渲染视图。

closeCamera

closeCamera

void closeCamera ()

关闭摄像头

switchCamera

switchCamera

void switchCamera (TXIoTCamera camera)

切换摄像头（前后置）

参数	描述
camera	TXIoTCamera；摄像头（前置/后置）。

selectAudioPlaybackDevice

selectAudioPlaybackDevice

void selectAudioPlaybackDevice (TXIoTAudioPlaybackDevice device)

选择音频播放设备（扬声器/听筒）

参数	描述
device	TXIoTAudioPlaybackDevice；音频播放设备（扬声器/听筒）。

通话事件回调

TXIoTCallListener

回调方法	说明

`onError(TXIoTErrorCode code, String msg)`	通话发生错误。
`onCallBegin(TXIoTCallMediaType mediaType)`	通话开始。
`onCallEnd(TXIoTCallMediaType mediaType, TXIoTCallEndReason reason)`	通话结束，携带结束原因。
`onCallRejected(TXIoTCallUser callUser)`	通话被对方拒接。
`onCallNoResponse(TXIoTCallUser callUser)`	对方无应答。
`onCallLineBusy(TXIoTCallUser callUser)`	对方占线。
`onCallUserOffline(TXIoTCallUser callUser)`	对方离线。
`onCallUserAudioAvailable(TXIoTCallUser callUser, boolean available)`	远端用户音频状态发生变化。
`onCallUserVideoAvailable(TXIoTCallUser callUser, boolean available)`	远端用户视频状态发生变化。
`onCallNetworkQualityChanged(TXIoTQuality localQuality, List<TXIoTQuality> remoteQualityList)`	网络质量变化通知。

通话相关数据结构与枚举

TXIoTCallMediaType

TXIoTCallMediaType

枚举	说明
AUDIO	音频通话。
VIDEO	视频通话。

TXIoTCallEndReason

TXIoTCallEndReason

枚举	说明
UNKNOWN	未知原因。
LOCAL_HANGUP	本地挂断。
REMOTE_HANGUP	远端挂断。
CALL_PERMISSION_DENIED	呼叫权限被拒绝。
NETWORK_ERROR	网络错误。

TXIoTAudioPlaybackDevice

TXIoTAudioPlaybackDevice

枚举	说明
SPEAKER_PHONE	扬声器（外放）。
EAR_PIECE	听筒。

TXIoTCamera

TXIoTCamera

枚举	说明
FRONT	前置摄像头。
BACK	后置摄像头。

TXIoTNetworkQuality

TXIoTNetworkQuality

枚举	说明
EXCELLENT	极佳
GOOD	良好
POOR	较差
BAD	差
VERY_BAD	很差
DOWN	网络不可用

TXIoTCallUser

TXIoTCallUser

字段	说明
deviceId	参与通话的设备 ID。

TXIoTQuality

TXIoTQuality

字段	说明
callUser	网络质量对应的通话用户。
quality	网络质量等级。

TXIoTMonitorSession

最近更新时间：2026-07-30 12:00:38

概述

TXIoTMonitorSession 负责拉取设备实时监控画面，包括开始/停止会话、远端拉流与清晰度切换、与设备进行对讲、云台控制、截图与本地录制。TXIoTMonitorSession 通过 TXIoTEngine.getMonitorSession 获取实例，需在登录成功后调用。

> **说明：**
> 回调模型：所有 Listener 回调在主线程触发，公共类型与错误码定义见 TXIoTEngineDef 。

方法列表

方法签名	描述
addListener	添加实时监控事件监听器。
removeListener	移除实时监控事件监听器。
startSession	开启实时监控。
stopSession	停止实时监控。
startRemoteView	开始渲染远端视频流。
stopRemoteView	停止渲染远端视频流。
switchRemoteStream	切换远端视频流类型（高清/标清）。
muteRemoteAudio	静音/取消静音单路远端音频。
muteAllRemoteAudio	静音/取消静音所有远端音频。
startLocalAudio	开启本地音频采集和推送（用于实现对讲功能）。
stopLocalAudio	关闭本地音频采集和推送。
muteLocalAudio	静音/取消静音本地音频。
startLocalVideo	开启本地视频采集和推送。
stopLocalVideo	关闭本地视频采集和推送。
sendPTZCommand	发送云台控制指令。

takeSnapshot	对当前正在观看的通道进行截图。
startLocalRecording	开始对当前正在观看的通道进行录像。
stopLocalRecording	停止本地录像。

addListener

addListener
void addListener (TXIoTMonitorSessionListener listener)

添加实时监控事件监听器

参数	描述
listener	`TXIoTMonitorSessionListener`；监听器实例。

removeListener

removeListener
void removeListener (TXIoTMonitorSessionListener listener)

移除实时监控事件监听器

参数	描述
listener	`TXIoTMonitorSessionListener`；监听器实例。

startSession

startSession
void startSession (TXIoTDeviceId deviceId)

开启实时监控

参数	描述
deviceId	TXIoTDeviceId；设备标识（含 productId / deviceName）。

stopSession

stopSession
void stopSession ()

停止实时监控

startRemoteView

startRemoteView
void startRemoteView (int channelId, TXIoTStreamType streamType, TXCloud

开始渲染远端视频流

参数	描述
channelId	`int`；通道 ID。
streamType	TXIoTStreamType；视频流类型（高清/标清）。
view	`TXCloudVideoView`；视频渲染视图。

stopRemoteView

stopRemoteView
void stopRemoteView (int channelId)

停止渲染远端视频流

参数	描述
channelId	`int`；通道 ID。

switchRemoteStream

void switchRemoteStream (int channelId, TXIoTStreamType streamType)

切换远端视频流类型（高清/标清）

参数	描述
channelId	`int`；通道 ID。
streamType	TXIoTStreamType；视频流类型（高清/标清）。

muteRemoteAudio

void muteRemoteAudio (int channelId, boolean mute)

静音/取消静音单路远端音频

参数	描述
channelId	`int`；通道 ID。
mute	`boolean`；是否静音。

muteAllRemoteAudio

muteAllRemoteAudio

void muteAllRemoteAudio (boolean mute)

静音/取消静音所有远端音频

参数	描述
mute	`boolean`；是否静音。

startLocalAudio

startLocalAudio

void startLocalAudio ()

开启本地音频采集和推送（用于实现对讲功能）

stopLocalAudio

stopLocalAudio

void stopLocalAudio ()

停止本地音频采集和推送

muteLocalAudio

muteLocalAudio

void muteLocalAudio (boolean mute)

静音/取消静音本地音频

参数	描述
mute	`boolean`；是否静音。

startLocalVideo

startLocalVideo
void startLocalVideo (TXCloudVideoView view, TXIoTVideoEncoderParams videoEncoderParams)

开启本地视频采集和推送

参数	描述
view	`TXCloudVideoView`；视频渲染视图。
videoEncoderParams	TXIoTVideoEncoderParams；视频编码参数。

stopLocalVideo

stopLocalVideo
void stopLocalVideo ()

停止本地视频采集和推送

sendPTZCommand

sendPTZCommand
void sendPTZCommand (int channelId, TXIoTPTZCommand command, int spee

发送云台控制指令

参数	描述
channelId	`int`；通道 ID。
command	TXIoTPTZCommand；云台控制指令。

speed	`int`；云台运动速度。

takeSnapshot

```
void takeSnapshot (int channelId)
```

对当前正在观看的通道进行截图

参数	描述
channelId	`int`；通道 ID。

startLocalRecording

```
void startLocalRecording (int channelId, TXIoTLocalRecordingParams params)
```

开始对当前正在观看的通道进行录像

参数	描述
channelId	`int`；通道 ID。
params	TXIoTLocalRecordingParams；本地录像参数。

stopLocalRecording

```
void stopLocalRecording (int channelId)
```

停止本地录像

参数	描述
channelId	`int`；通道 ID。

监控事件回调

TXIoTMonitorSessionListener

回调方法	说明
`onError(int channelId, TXIoTErrorCode errCode, String errMsg)`	监控发生错误。
`onSessionEstablished()`	监控会话建立成功。
`onSessionReconnecting()`	会话重连中。
`onSessionRecovery()`	会话恢复成功。
`onRemoteStreamAvailable(int channelId, boolean available)`	远端流可用性变化。
`onRenderFirstFrame(int channelId)`	首帧渲染完成。
`onPlayStateChanged(int channelId, TXIoTPlayState state)`	播放状态变化。
`onSnapshotComplete(int channelId, Bitmap image, TXIoTErrorCode errCode)`	截图完成，成功时 image 非 null。
`onLocalRecordBegin(int channelId, TXIoTErrorCode errCode, String storagePath)`	本地录像开始。
`onLocalRecording(int channelId, long durationMs, String storagePath)`	本地录像进行中，周期性回调时长。
`onLocalRecordComplete(int channelId, TXIoTErrorCode errCode, String storagePath)`	本地录像完成。

监控相关数据结构与枚举

TXIoTStreamType

TXIoTStreamType

枚举	说明
HD	高清
SD	标清

TXIoTPTZCommand

TXIoTPTZCommand

枚举	说明
UP	上
DOWN	下
LEFT	左
RIGHT	右
ZOOM_IN	变焦放大
ZOOM_OUT	变焦缩小
STOP	停止

TXIoTPlayState

TXIoTPlayState

枚举	说明

PLAYING	正在播放
LOADING	加载中
STOPPED	已停止

TXIoTVideoResolution

TXIoTVideoResolution

枚举	说明
RESOLUTION_640_360	横屏 640×360
RESOLUTION_960_540	横屏 960×540
RESOLUTION_1280_720	横屏 1280×720
RESOLUTION_1920_1080	横屏 1920×1080
RESOLUTION_360_640	竖屏 360×640
RESOLUTION_540_960	竖屏 540×960
RESOLUTION_720_1280	竖屏 720×1280
RESOLUTION_1080_1920	竖屏 1080×1920

TXIoTVideoEncoderParams

TXIoTVideoEncoderParams

字段	说明
resolution	视频编码分辨率。

TXIoTLocalRecordingParams

TXIoTLocalRecordingParams

字段	说明
filePath	录像文件保存路径。

iOS
TXIoTEngine

最近更新时间：2026-07-30 12:00:38

概述

`TXIoTEngine` 是整个 SDK 的入口，通过 `+(instancetype)getInstance` 获取单例，使用前须先调用 `login` 完成身份验证。

> **说明：**
> 线程模型：SDK 接口可在任意线程调用；回调均在主线程（UI 线程）触发，调用方可在回调中直接操作 UI。

SDK 模块一览：

模块	说明
TXIoTEngine 引擎管理	登录、获取各个业务对象。
TXIoTFamilyManager 家庭与房间管理	家庭、房间、成员管理。
TXIoTDeviceManager 设备管理	设备绑定/解绑、分享、指令下发、属性查询。
TXIoTCallSession 音视频通话	音视频通话。
TXIoTMonitorSession 监控直播	监控直播（拉流/对讲/云台/截图/录像）。
公共类型与枚举	错误码、通用枚举、公共数据结构、回调接口。

方法列表

方法签名	描述
getInstance	获取引擎单例。
addDelegate: / removeDelegate:	添加/移除引擎事件代理。
login:userId:userSignature:	使用 appKey、userId、用户签名登录。
logout	登出当前账号。

getLoginUserInfo	获取当前登录用户信息。
getFamilyManager	获取家庭管理器，未登录时返回 null。
getDeviceManager	获取设备管理器，未登录时返回 null。
getCallSession	获取音视频通话会话，未登录时返回 null。
getMonitorSession	获取实时监控会话，未登录时返回 null。

getInstance

getInstance

TXIoTEngine 实例 getInstance ()

获取引擎单例

返回值	描述
TXIoTEngine 实例	SDK 引擎单例对象。

addDelegate:

addDelegate:

void addDelegate: (id<TXIoTEngineDelegate> delegate)

添加引擎事件代理对象

参数	描述
delegate	`id<TXIoTEngineDelegate>`；引擎事件代理对象。

removeDelegate:

removeDelegate:

void removeDelegate: (id<TXIoTEngineDelegate> delegate)

移除引擎事件代理对象

参数	描述
delegate	`id<TXIoTEngineDelegate>`；要移除的引擎事件代理对象。

login:userId:userSignature:

login:userId:userSignature:
void login:userId:userSignature: (NSString* appKey, NSString* userId, TXIoTUserSignat userSignature)

使用 appKey、userId、用户签名进行登录

参数	描述
appKey	`NSString*`；应用密钥。
userId	`NSString*`；用户 ID。支持字母、数字、下划线，长度不超过 32 字节。
userSignature	TXIoTUserSignature*；用户签名对象，见 TXIoTUserSignature 。

logout

logout
void logout ()

登出当前账号

getLoginUserInfo

getLoginUserInfo
TXIoTUserInfo* ()

获取当前已登录的用户信息

返回值	描述
TXIoTUserInfo *	已登录返回用户信息对象；未登录返回 nil。类型定义见 TXIoTUserInfo 。

getFamilyManager

getFamilyManager
TXIoTFamilyManager* getFamilyManager ()

获取家庭管理器

返回值	描述
TXIoTFamilyManager*	已登录返回家庭管理器实例；未登录返回 nil。详见 TXIoTFamilyManager 。

getDeviceManager

getDeviceManager
TXIoTDeviceManager* getDeviceManager ()

获取设备管理器

返回值	描述
TXIoTDeviceManager*	已登录返回设备管理器实例；未登录返回 nil。详见 TXIoTDeviceManager 。

getCallSession

getCallSession
TXIoTCallSession*
getCallSession
()

获取音视频通话会话

返回值	描述
TXIoTCallSession*	已登录返回通话会话实例；未登录返回 nil。详见 TXIoTCallSession 。

getMonitorSession

getMonitorSession
TXIoTMonitorSession*
getMonitorSession
()

获取实时监控会话

返回值	描述
TXIoTMonitorSession*	已登录返回监控会话实例；未登录返回 nil。详见 TXIoTMonitorSession 。

引擎事件回调

`TXIoTEngineDelegate` 协议定义引擎层事件回调，所有方法均为 `@optional` 。

TXIoTEngineDelegate

回调	说明
onLoginSuccess	登录成功。
onLoginFailure:errMsg:	登录失败，返回 TXIoTErrorCode 和错误信息。
onLogout	登出完成。

onUserSignatureExpired	用户签名已过期，需重新登录。
onReceivePushMessage:	收到推送消息（如设备上下线状态变更）。

引擎相关数据结构与枚举

TXIoTPushMessage

TXIoTPushMessage
推送消息，通过 `onReceivePushMessage:` 回调接收。

字段	说明
messageTime	推送消息时间戳（Unix 秒级时间戳）。
type	推送消息类型。
subType	推送消息子类型。
deviceId	关联的设备 ID。

TXIoTUserSignature

TXIoTUserSignature
用户签名，用于登录认证。

字段	说明
requestId	请求 ID，由服务端签发。
timestamp	签名的 Unix 秒级时间戳。
nonce	签名随机数。
signature	签名字符串，由服务端签发。

TXIoTPushMessageType

TXIoTPushMessageType
推送消息类型枚举。

枚举	取值
TXIoTPushMessageTypeStatusChange	0

TXIoTPushMessageSubType

TXIoTPushMessageSubType
推送消息子类型枚举。

枚举	取值
TXIoTPushMessageSubTypeOnline	0
TXIoTPushMessageSubTypeOffline	1

TXIoTEngineDef

最近更新时间：2026-07-30 12:00:38

概述

本文档描述 IoT App SDK 在 iOS 端的公共类型与枚举定义，包含错误码、通用枚举、公共数据结构与通用回调等，定义于 `TXIoTEngineDef.h` 。

> **说明：**
> `TXIoTEngineDef.h` 是各模块共用的基础类型定义头文件，各模块文档见：
> - TXIoTEngine：引擎管理。
> - TXIoTFamilyManager：家庭与房间管理。
> - TXIoTDeviceManager：设备管理。
> - TXIoTCallSession：音视频通话。
> - TXIoTMonitorSession：实时监控。

错误码 — TXIoTErrorCode

TXIoTErrorCode

错误码	取值	
TXIoTErrorCodeSuccess	0	成功。
TXIoTErrorCodeFailed	-1	通用失败。
TXIoTErrorCodeInvalidParameter	-2	参数非法。
TXIoTErrorCodeInvalidAccessToken	-3	Access Token 无效。
TXIoTErrorCodeRateLimited	-4	请求频率超限。
TXIoTErrorCodeUnauthorizedOperation	-5	未授权操作。
TXIoTErrorCodeDeviceBound	-1000	设备已绑定。
TXIoTErrorCodeBindTokenNotE	-1001	绑定令牌不存在。

xist		
TXIoTErrorCodeBindTokenIsExpired	-1002	绑定令牌已过期。
TXIoTErrorCodeBindTokenNoPairWithDevice	-1003	绑定令牌与设备不匹配。
TXIoTErrorCodeInvalidMessageId	-1004	消息 ID 无效。
TXIoTErrorCodeFamilyDeviceLimitExceeded	-1005	家庭设备数量超限。
TXIoTErrorCodeCanNotBindSameFamily	-1006	不能绑定到同一家庭。
TXIoTErrorCodeFamilyNotExist	-1007	家庭不存在。
TXIoTErrorCodeProductNotExist	-1008	产品不存在。
TXIoTErrorCodeDeviceNotExist	-1009	设备不存在。
TXIoTErrorCodeDeviceOffline	-1010	设备离线。
TXIoTErrorCodeCameraStartFail	-2000	摄像头启动失败。
TXIoTErrorCodeCameraNotAuthorized	-2001	摄像头权限未授权。
TXIoTErrorCodeCameraOccupy	-2002	摄像头被占用。
TXIoTErrorCodeMicStartFail	-2003	麦克风启动失败。
TXIoTErrorCodeMicNotAuthorized	-2004	麦克风权限未授权。
TXIoTErrorCodeMicOccupy	-2005	麦克风被占用。
TXIoTErrorCodeSpeakerStartFail	-2006	扬声器启动失败。
TXIoTErrorCodeRecordDurationTooShort	-2007	录制时长过短。
TXIoTErrorCodeFilePathUnwritable	-2008	文件路径不可写。

TXIoTErrorCodeFilePathAlready Exist	-2009	文件路径已存在。
TXIoTErrorCodeDeviceSwitchTo VoIP	-2010	设备切换到 VOIP 模式。

通用枚举

TXIoTFamilyRole

TXIoTFamilyRole
家庭角色。

枚举	取值
TXIoTFamilyRoleUnknown	0
TXIoTFamilyRoleMember	1
TXIoTFamilyRoleAdmin	2

公共数据结构

TXIoTDeviceId

TXIoTDeviceId
设备 ID，用于唯一标识一个设备。在 TXIoTEngine 、TXIoTDeviceManager 、TXIoTCallSession 、TXIoTMonitorSession 中使用。

字段	说明
productId	产品 ID。
deviceName	设备名称。

TXIoTDeviceStatus

TXIoTDeviceStatus
设备在线状态。

字段	说明
isOnline	是否在线。
onlineDuration	累计在线时长（单位：秒）。
bringOnlineTime	上线时间戳（Unix 秒级时间戳）。

TXIoTDeviceInfo

TXIoTDeviceInfo
设备信息。在 TXIoTDeviceManager 中使用。

字段	说明
deviceId	设备 ID。
aliasName	设备别名。
familyId	所属家庭 ID。
roomId	所属房间 ID。
iconUrl	设备图标 URL。
createTime	创建时间（Unix 秒级时间戳）。
updateTime	更新时间（Unix 秒级时间戳）。
status	设备在线状态。

TXIoTFamilyInfo

TXIoTFamilyInfo
家庭信息。在 TXIoTFamilyManager 中使用。

字段	说明
familyId	家庭 ID。
name	家庭名称。
createTime	创建时间（Unix 秒级时间戳）。
updateTime	更新时间（Unix 秒级时间戳）。
role	当前用户在该家庭中的角色。

TXIoTRoomInfo

TXIoTRoomInfo
房间信息。在 TXIoTFamilyManager 中使用。

字段	说明
roomId	房间 ID。
name	房间名称。
familyId	所属家庭 ID。
deviceCount	房间内设备数量。
createTime	创建时间（Unix 秒级时间戳）。
updateTime	更新时间（Unix 秒级时间戳）。

TXIoTUserInfo

TXIoTUserInfo
用户信息。在 TXIoTEngine、TXIoTFamilyManager、TXIoTDeviceManager 中使用。

字段	说明
userId	用户 ID。
nickName	用户昵称。
avatarUrl	用户头像 URL。
role	用户角色。
deviceBindTime	设备绑定时间（Unix 秒级时间戳）。

TXIoTPageResult<T>

TXIoTPageResult<T>
分页查询结果泛型类。在 TXIoTDeviceManager 中使用。

字段	说明
dataList	当前页数据列表。
nextPageToken	下一页游标；首次查询传空字符串，取结果中返回值用于获取下一页；为空字符串表示已到最后一页。

通用回调

TXIoTCallback<ResultType>

TXIoTCallback<ResultType>
有返回值的异步回调。

```
@interface TXIoTCallback<__covariant ResultType> : NSObject
@property(nonatomic, copy) void (^onSuccess)(ResultType _Nullable
result);
@property(nonatomic, copy) void (^onError)(TXIoTErrorCode errorCode,
NSString* _Nullable errorMessage);
```

```
@end
```

TXIoTVoidCallback

TXIoTVoidCallback
无返回值的异步回调。

```
@interface TXIoTVoidCallback : NSObject
@property(nonatomic, copy) void (^onSuccess)(void);
@property(nonatomic, copy) void (^onError)(TXIoTErrorCode errorCode,
NSString* _Nullable errorMessage);
@end
```

TXIoTFamilyManager

最近更新时间：2026-07-30 12:00:38

概述

TXIoTFamilyManager 负责家庭的创建/删除/更新、房间的创建/删除/重命名，以及家庭成员的邀请与管理。TXIoTFamilyManager 通过 TXIoTEngine.getFamilyManager 获取实例，需在 TXIoTEngine 登录成功后调用。

> **说明：**
> 回调模型：通过 TXIoTCallback / TXIoTVoidCallback 回调返回结果的接口，回调在主线程触发。

方法列表

方法签名	描述
createFamily:callback:	创建家庭。
deleteFamily:callback:	删除家庭。
getFamilyList:	获取家庭列表。
updateFamilyInfo:callback:	更新家庭信息。
createRoom:name:callback:	在家庭下创建房间。
deleteRoom:roomId:callback:	删除房间。
setRoomName:roomId:name:callback:	设置房间名称。
getRoomList:callback:	获取家庭下的房间列表。
createFamilyInviteToken:callback:	创建家庭邀请令牌。
joinFamilyAsMember:callback:	通过邀请令牌以成员身份加入家庭。
removeMemberFromFamily:userId:callback:	将成员移出家庭。
getMemberList:callback:	获取家庭成员列表。

createFamily:callback:

createFamily:callback:
void createFamily:callback: (NSString* name, TXIoTCallback<TXIoTFamilyInfo> cal

创建家庭

参数	描述
name	`NSString*`；家庭名称。
callback	TXIoTCallback<TXIoTFamilyInfo>；异步回调，通过 onSuccess / onError 通知结果。

deleteFamily:callback:

deleteFamily:callback:
void deleteFamily:callback: (NSString* familyId, TXIoTVoidCallback* callback)

删除家庭

参数	描述
familyId	`NSString*`；家庭 ID。
callback	TXIoTVoidCallback*；异步回调，通过 onSuccess / onError 通知结果。

getFamilyList:

getFamilyList:
void getFamilyList: (TXIoTCallback<NSArray<TXIoTFamilyInfo*>> callback

获取家庭列表

参数	描述

callback	TXIoTCallback<NSArray<TXIoTFamilyInfo *>>；异步回调，通过 onSuccess / onError 通知结果。

updateFamilyInfo:callback:

```
void updateFamilyInfo:callback:    (TXIoTFamilyInfo* newFamilyInfo, TXIoTVoidCallback*
```

更新家庭信息

参数	描述
newFamilyInfo	TXIoTFamilyInfo *；新的家庭信息对象。
callback	TXIoTVoidCallback *；异步回调，通过 onSuccess / onError 通知结果。

createRoom:name:callback:

```
void createRoom:name:callback:    (NSString* familyId, NSString* name, TXIoTCallback<T
                                  callback)
```

在家庭下创建房间

参数	描述
familyId	`NSString*`；家庭 ID。
name	`NSString*`；房间名称。
callback	TXIoTCallback<TXIoTRoomInfo >；异步回调，通过 onSuccess / onError 通知结果。

deleteRoom:roomId:callback:

deleteRoom:roomId:callback:

```
void
deleteRoom:roomId:callback:
(NSString* familyId, NSString* roomId, TXIoTVoidCallba
```

删除房间

参数	描述
familyId	`NSString*`；家庭 ID。
roomId	`NSString*`；房间 ID。
callback	TXIoTVoidCallback*；异步回调，通过 onSuccess / onError 通知结果。

setRoomName:roomId:name:callback:

setRoomName:roomId:name:callback:

```
void
setRoomName:roomId:name:cal
lback:
(NSString* familyId, NSString* roomId, NSString* name
TXIoTVoidCallback* callback)
```

设置房间名称

参数	描述
familyId	`NSString*`；家庭 ID。
roomId	`NSString*`；房间 ID。
name	`NSString*`；新名称。
callback	TXIoTVoidCallback*；异步回调，通过 onSuccess / onError 通知结果。

getRoomList:callback:

getRoomList:callback:
void getRoomList:callback: (NSString* familyId, TXIoTCallback<NSArray<TXIoTRo
callback)

获取家庭下的房间列表

参数	描述
familyId	`NSString*`；家庭 ID。
callback	TXIoTCallback<NSArray<TXIoTRoomInfo*>>；异步回调，通过 onSuccess / onError 通知结果。

createFamilyInviteToken:callback:

createFamilyInviteToken:callback:
void createFamilyInviteToken:callback: (NSString* familyId, TXIoTCallback<NSString*>* callba

创建家庭邀请令牌

参数	描述
familyId	`NSString*`；家庭 ID。
callback	TXIoTCallback<NSString*>*；异步回调，通过 onSuccess / onError 通知结果。

joinFamilyAsMember:callback:

- void joinFamilyAsMember:callback:(NSString* inviteToken, TXIoTVoidCallback* callback)

joinFamilyAsMember:callback:

通过邀请令牌以成员身份加入家庭

参数	描述
inviteToken	`NSString*`；邀请令牌。
callback	TXIoTVoidCallback*；异步回调，通过 onSuccess / onError 通知结果。

removeMemberFromFamily:userId:callback:

- void removeMemberFromFamily:userId:callback:(NSString* familyId, NSString* userId, TXIoTVoidCallba

removeMemberFromFamily:userId:callback:

将成员移出家庭

参数	描述
familyId	`NSString*`；家庭 ID。
userId	`NSString*`；用户 ID。
callback	TXIoTVoidCallback*；异步回调，通过 onSuccess / onError 通知结果。

getMemberList:callback:

getMemberList:callback:
void getMemberList:callback:

(NSString* familyId, TXIoTCallback<NSArray<TXIoTUs
callback)

获取家庭成员列表

参数	描述
familyId	`NSString*`；家庭 ID。
callback	TXIoTCallback<NSArray<TXIoTUserInfo*>>；异步回调，通过 onSuccess / onError 通知结果。

TXIoTDeviceManager

最近更新时间：2026-07-30 12:00:38

概述

TXIoTDeviceManager 负责设备的绑定/解绑、房间归属、设备分享、指令下发、属性查询与别名修改等。TXIoTDeviceManager 通过 TXIoTEngine.getDeviceManager 获取实例，需在 TXIoTEngine 登录成功后调用。

> **说明：**
> 回调模型：所有接口为异步操作，通过 TXIoTCallback / TXIoTVoidCallback 回调返回结果，回调在主线程触发。公共类型定义见 TXIoTEngineDef。

方法列表

方法签名	描述
bindDevice:deviceBindSignature:callback:	绑定设备到指定家庭。
unbindDevice:deviceId:callback:	从家庭解绑设备。
addDeviceToRoom:familyId:roomId:callback:	将设备加入房间。
removeDeviceFromRoom:familyId:callback:	将设备移出房间。
getDeviceList:nextPageToken:callback:	获取家庭下的设备列表（支持分页）。
createDeviceSharingToken:deviceId:callback:	创建设备分享令牌。
bindDeviceSharedWithMe:shareToken:callback:	绑定他人分享给我的设备。
unbindDeviceSharedWithMe:callback:	解绑他人分享给我的设备。
getDeviceSharedUsers:callback:	获取设备被分享的用户列表。

removeDeviceSharedUser:userId:callback:	移除设备的某个被分享用户。
getDeviceListSharedWithMe:callback:	获取分享给我的设备列表（支持分页）。
sendCommand:jsonData:callback:	向设备下发控制指令。
getProperties:callback:	获取设备属性。
modifyAliasName:aliasName:callback:	修改设备别名。

bindDevice:deviceBindSignature:callback:

bindDevice:deviceBindSignature:callback:

void bindDevice:deviceBindSignature:callback: (NSString* familyId, NSString* deviceBindSignature, TXIoTCallback<TXIoTDeviceInfo> callback)

绑定设备到指定家庭

参数	描述
familyId	`NSString*`；家庭 ID。
deviceBindSignature	`NSString*`；设备绑定签名。
callback	TXIoTCallback<TXIoTDeviceInfo>；异步回调，通过 onSuccess / onError 通知结果，成功时返回设备信息。

unbindDevice:deviceId:callback:

unbindDevice:deviceId:callback:

void (NSString* familyId, TXIoTDeviceId* deviceId, TXIoTVoi

从家庭解绑设备

参数	描述
familyId	`NSString*`；家庭 ID。
deviceId	TXIoTDeviceId*；设备 ID。
callback	TXIoTVoidCallback*；异步回调，通过 onSuccess / onError 通知结果。

addDeviceToRoom:familyId:roomId:callback:

addDeviceToRoom:familyId:roomId:callback:

```
void addDeviceToRoom:familyId:roomId:callback:(TXIoTDeviceId* deviceId, NSString* familyId, NSString
TXIoTVoidCallback* callback)
```

将设备加入房间

参数	描述
deviceId	TXIoTDeviceId*；设备 ID。
familyId	`NSString*`；家庭 ID。
roomId	`NSString*`；房间 ID。
callback	TXIoTVoidCallback*；异步回调，通过 onSuccess / onError 通知结果。

removeDeviceFromRoom:familyId:callback:

removeDeviceFromRoom:familyId:callback:

```
void
removeDeviceFromRoom:familyId:callback:
(TXIoTDeviceId* deviceId, NSString* familyId, TXIoTVoi
callback)
```

将设备移出房间

参数	描述
deviceId	TXIoTDeviceId*；设备 ID。
familyId	`NSString*`；家庭 ID。
callback	TXIoTVoidCallback*；异步回调，通过 onSuccess / onError 通知结果。

getDeviceList:nextPageToken:callback:

getDeviceList:nextPageToken:callback:

```
void
getDeviceList:nextPageToken:callback:
(NSString* familyId, NSString* _Nullable nextPageToke
TXIoTCallback<TXIoTPageResult<TXIoTDeviceInfo>>*
```

获取家庭下的设备列表（支持分页）

参数	描述
familyId	`NSString*`；家庭 ID。
nextPageToken	`NSString* _Nullable`；分页标记，首页传空字符串。
callback	TXIoTCallback<TXIoTPageResult<TXIoTDeviceInfo>>*；异步回调，通过 onSuccess / onError 通知结果。

createDeviceSharingToken:deviceId:callback:

createDeviceSharingToken:deviceId:callback:

```
void createDeviceSharingToken:deviceId:callback:(NSString* familyId, TXIoTDeviceId* deviceId, TXIoTCallback<NSString*>* callback)
```

创建设备分享令牌

参数	描述
familyId	`NSString*`；家庭 ID。
deviceId	TXIoTDeviceId*；设备 ID。
callback	TXIoTCallback<NSString*>*；异步回调，通过 onSuccess / onError 通知结果，成功时返回分享令牌字符串。

bindDeviceSharedWithMe:shareToken:callback:

bindDeviceSharedWithMe:shareToken:callback:

```
void bindDeviceSharedWithMe:shareToken:callback:(TXIoTDeviceId* deviceId, NSString* shareToken, TXIo
callback)
```

绑定他人分享给我的设备

参数	描述
deviceId	TXIoTDeviceId*；设备 ID。
shareToken	`NSString*`；分享令牌。
callback	TXIoTVoidCallback*；异步回调，通过 onSuccess / onError 通知结果。

unbindDeviceSharedWithMe:callback:

unbindDeviceSharedWithMe:callback:

```
void
unbindDeviceSharedWithMe:call      (TXIoTDeviceId* deviceId, TXIoTVoidCallback* callback
back:
```

解绑他人分享给我的设备

参数	描述
deviceId	TXIoTDeviceId*；设备 ID。
callback	TXIoTVoidCallback*；异步回调，通过 onSuccess / onError 通知结果。

getDeviceSharedUsers:callback:

getDeviceSharedUsers:callback:

```
void
getDeviceSharedUsers:callback      (TXIoTDeviceId* deviceId, TXIoTCallback<NSArray<TX
:                                  callback)
```

获取设备被分享的用户列表

参数	描述
deviceId	TXIoTDeviceId*；设备 ID。
callback	TXIoTCallback<NSArray<TXIoTUserInfo*>>；异步回调，通过 onSuccess / onError 通知结果，成功时返回用户列表。

removeDeviceSharedUser:userId:callback:

removeDeviceSharedUser:userId:callback:

```
void
removeDeviceSharedUser:userId:callback:
```

(TXIoTDeviceId* deviceId, NSString* userId, TXIoTVoidCallback)

移除设备的某个被分享用户

参数	描述
deviceId	TXIoTDeviceId*；设备 ID。
userId	`NSString\*`；用户 ID。
callback	TXIoTVoidCallback*；异步回调，通过 onSuccess / onError 通知结果。

getDeviceListSharedWithMe:callback:

getDeviceListSharedWithMe:callback:

```
void
getDeviceListSharedWithMe:callback:
```

(NSString* _Nullable nextPageToken, TXIoTCallback<TXIoTPageResult<TXIoTDeviceInfo>>*

获取分享给我的设备列表（支持分页）

参数	描述
nextPageToken	`NSString\* _Nullable`；分页标记，首页传 nil。
callback	TXIoTCallback<TXIoTPageResult<TXIoTDeviceInfo>>*；异步回调，通过 onSuccess / onError 通知结果，成功时返回分页结果。

sendCommand:jsonData:callback:

sendCommand:jsonData:callback:

void sendCommand:jsonData:callback: (TXIoTDeviceId* deviceId, NSString* jsonData, TXIoTCallback<NSString*>* callback)

向设备下发控制指令

参数	描述
deviceId	TXIoTDeviceId*；设备 ID。
jsonData	`NSString*`；JSON 格式的控制指令数据。
callback	TXIoTCallback<NSString*>*；异步回调，通过 onSuccess / onError 通知结果，成功时返回设备响应数据。

getProperties:callback:

getProperties:callback:

void getProperties:callback: (TXIoTDeviceId* deviceId, TXIoTCallback<NSString*>*

获取设备属性

参数	描述
deviceId	TXIoTDeviceId*；设备 ID。
callback	TXIoTCallback<NSString*>*；异步回调，通过 onSuccess / onError 通知结果，成功时返回 JSON 格式的设备属性。

modifyAliasName:aliasName:callback:

modifyAliasName:aliasName:callback:

void modifyAliasName:aliasName:callback:	(TXIoTDeviceId* deviceId, NSString* aliasName, TXIoT callback)

修改设备别名

参数	描述
deviceId	TXIoTDeviceId*；设备 ID。
aliasName	`NSString*`；新的别名。
callback	TXIoTVoidCallback*；异步回调，通过 onSuccess / onError 通知结果。

TXIoTCallSession

最近更新时间：2026-07-30 12:00:38

概述

TXIoTCallSession 负责与设备之间的音视频通话，包括发起呼叫、挂断、远端画面渲染、麦克风/摄像头控制及音频输出设备选择。TXIoTCallSession 通过 TXIoTEngine.getCallSession 获取实例，需在 TXIoTEngine 登录成功后调用。

> **说明：**
> 回调模型：通话事件通过 `TXIoTCallDelegate` 协议回调，所有 Delegate 回调在主线程触发。公共类型与错误码定义见 TXIoTEngineDef。

方法列表

方法签名	描述
addDelegate:	添加通话事件代理。
removeDelegate:	移除通话事件代理。
callDevice:callType:	向设备发起音视频通话。
hangup	挂断当前通话。
startRemoteView:view:	开始远端画面渲染。
stopRemoteView:	停止远端画面渲染。
openMicrophone	打开麦克风。
closeMicrophone	关闭麦克风。
openCamera:view:	打开摄像头。
closeCamera	关闭摄像头。
switchCamera:	切换摄像头。
selectAudioPlaybackDevice:	选择音频播放设备。

addDelegate:

```
void addDelegate:(id<TXIoTCallDelegate> delegate)
```

添加通话事件代理

参数	描述
delegate	`id<TXIoTCallDelegate>`；通话事件代理对象。

removeDelegate:

```
void removeDelegate:(id<TXIoTCallDelegate> delegate)
```

移除通话事件代理

参数	描述
delegate	`id<TXIoTCallDelegate>`；要移除的通话事件代理对象。

callDevice:callType:

```
void callDevice:(TXIoTDeviceId* deviceId, TXIoTCallMediaType callTyp
```

向设备发起音视频通话

参数	描述
deviceId	TXIoTDeviceId*；目标设备 ID。

callType	TXIoTCallMediaType；通话类型：音频或视频。

hangup

hangup

```
void hangup()
```

挂断当前通话

startRemoteView:view:

startRemoteView:view:

```
void startRemoteView:view:(TXIoTCallUser* callUser, TXView* view)
```

开始渲染远端画面

参数	描述
callUser	TXIoTCallUser*；通话用户对象。
view	`TXView*`；用于渲染远端画面的视图。

stopRemoteView:

stopRemoteView:

```
void stopRemoteView:(TXIoTCallUser* callUser)
```

停止远端画面渲染

参数	描述

callUser	TXIoTCallUser*；参与本次通话的用户对象。

openMicrophone

```
void openMicrophone ()
```

打开麦克风

closeMicrophone

```
void closeMicrophone ()
```

关闭麦克风

openCamera:view:

```
void openCamera:view: (TXIoTCamera camera, TXView* view)
```

打开摄像头

参数	描述
camera	TXIoTCamera；摄像头类型：前置或后置。
view	`TXView*`；用于渲染本地画面的视图。

closeCamera

closeCamera

void closeCamera ()

关闭摄像头

switchCamera:

switchCamera:

void switchCamera: (TXIoTCamera camera)

切换摄像头

参数	描述
camera	TXIoTCamera；目标摄像头类型：前置或后置。

selectAudioPlaybackDevice:

selectAudioPlaybackDevice:

void selectAudioPlaybackDevice: (TXIoTAudioPlaybackDevice device)

选择音频播放设备

参数	描述
device	TXIoTAudioPlaybackDevice；音频播放设备：扬声器或听筒。

通话事件回调

`TXIoTCallDelegate` 协议定义通话事件回调，所有方法均为 `@optional` 。

TXIoTCallDelegate

回调	说明
onError:errorMessage:	通话过程中发生错误，错误码见 TXIoTErrorCode 。
onCallBegin:	通话开始，返回通话类型（音频/视频）。
onCallEnd:reason:	通话结束，返回挂断原因。
onCallRejected:	对方拒绝接听。
onCallNoResponse:	对方无响应。
onCallLineBusy:	对方线路忙。
onCallUserOffline:	对方不在线。
onCallUserAudioAvailable:available:	远端音频可用性变化。
onCallUserVideoAvailable:available:	远端视频可用性变化。
onCallNetworkQualityChanged:remoteQualityList:	网络质量变化通知。

通话相关数据结构与枚举

TXIoTCallUser

TXIoTCallUser
参与通话的用户信息，用于标识通话对端。

字段	说明
deviceId	通话对端的设备 ID。

TXIoTQuality

TXIoTQuality
网络质量信息。

字段	说明
callUser	网络质量对应的通话用户。
quality	网络质量等级。

TXIoTCallMediaType

TXIoTCallMediaType
通话类型枚举。

枚举	取值	说明
TXIoTCallMediaTypeAudio	0	音频通话。
TXIoTCallMediaTypeVideo	1	视频通话。

TXIoTCallEndReason

TXIoTCallEndReason
通话结束原因枚举。

枚举	取值	说明
TXIoTCallEndReasonUnknown	0	未知原因。
TXIoTCallEndReasonLocalHangup	1	本地挂断。
TXIoTCallEndReasonRemoteHangup	2	远端挂断。

TXIoTCallEndReasonCallPermissionDenied	3	呼叫权限被拒绝。
TXIoTCallEndReasonNetworkError	4	网络错误。

TXIoTAudioPlaybackDevice

TXIoTAudioPlaybackDevice

枚举	取值	说明
TXIoTAudioPlaybackDeviceSpeakerphone	0	扬声器（外放）。
TXIoTAudioPlaybackDeviceEarpiece	1	听筒。

TXIoTCamera

TXIoTCamera

枚举	取值	说明
TXIoTCameraFront	0	前置摄像头。
TXIoTCameraBack	1	后置摄像头。

TXIoTNetworkQuality

TXIoTNetworkQuality

枚举	取值	说明

TXIoTNetworkQualityExcellent	0	极佳
TXIoTNetworkQualityGood	1	良好
TXIoTNetworkQualityPoor	2	较差
TXIoTNetworkQualityBad	3	差
TXIoTNetworkQualityVeryBad	4	很差
TXIoTNetworkQualityDown	5	网络不可用

TXIoTMonitorSession

最近更新时间：2026-07-30 12:00:38

概述

TXIoTMonitorSession 负责拉取设备实时监控画面，包括开始/停止会话、远端拉流与清晰度切换、与设备进行对讲、云台控制、截图与本地录制。TXIoTMonitorSession 通过 TXIoTEngine.getMonitorSession 获取实例，需在 TXIoTEngine 登录成功后调用。

> **说明：**
> 回调模型：监控事件通过 `TXIoTMonitorSessionDelegate` 协议回调，所有 Delegate 回调在主线程触发。公共类型与错误码定义见 TXIoTEngineDef。

方法列表

方法签名	描述
addDelegate:	添加实时监控事件代理。
removeDelegate:	移除实时监控事件代理。
startSession:	启动实时监控会话。
stopSession	停止实时监控会话。
startRemoteView:streamType:view:	开始远端拉流和渲染。
stopRemoteView:	停止远端拉流和渲染。
switchRemoteStream:streamType:	切换远端流的清晰度（高清/标清）。
muteRemoteAudio:mute:	静音/取消静音单路远端音频。
muteAllRemoteAudio:	静音/取消静音所有远端音频。
startLocalAudio	开始对讲（上行音频）。
stopLocalAudio	停止对讲。
muteLocalAudio:	静音本地音频。
startLocalVideo:videoEncoderP	开始本地视频采集和推送。

arams:	
stopLocalVideo	停止本地视频采集和推送。
sendPTZCommand:command:speed:	发送云台控制指令。
takeSnapshot:	对当前正在观看的通道进行截图。
startLocalRecording:params:	开始对当前正在观看的通道进行录制。
stopLocalRecording:	停止录制。

addDelegate:

addDelegate:
void addDelegate: (id<TXIoTMonitorSessionDelegate> delegate)

添加监控事件代理

参数	描述
delegate	`id<TXIoTMonitorSessionDelegate>`；监控事件代理对象。

removeDelegate:

removeDelegate:
void removeDelegate: (id<TXIoTMonitorSessionDelegate> delegate)

移除监控事件代理

参数	描述
delegate	`id<TXIoTMonitorSessionDelegate>`；要移除的监控事件代理对象。

startSession:

startSession:

void startSession: (TXIoTDeviceId* deviceId)

启动实时监控会话

会话建立结果通过 `onSessionEstablished` **回调通知**

参数	描述
deviceId	TXIoTDeviceId*；要监控的设备 ID。

stopSession

stopSession

void stopSession ()

停止监控会话

startRemoteView:streamType:view:

startRemoteView:streamType:view:

void startRemoteView:streamType:view: (NSInteger channelId, TXIoTStreamType streamType, T

开始远端拉流渲染

参数	描述
channelId	`NSInteger`；通道 ID。
streamType	TXIoTStreamType；流类型：高清或标清。
view	`TXView*`；用于渲染远端视频的视图。

stopRemoteView:

stopRemoteView:

```
void stopRemoteView:        (NSInteger channelId)
```

停止远端拉流渲染

参数	描述
channelId	`NSInteger`；通道 ID。

switchRemoteStream:streamType:

switchRemoteStream:streamType:

```
void
switchRemoteStream:streamType:        (NSInteger channelId, TXIoTStreamType streamType)
```

切换远端流的清晰度（高清/标清）

参数	描述
channelId	`NSInteger`；通道 ID。
streamType	TXIoTStreamType；目标流的清晰度。

muteRemoteAudio:mute:

muteRemoteAudio:mute:

```
void muteRemoteAudio:mute:        (NSInteger channelId, BOOL mute)
```

静音/取消静音单路远端音频

参数	描述
channelId	`NSInteger`；通道 ID。
mute	`BOOL`；YES 静音，NO 取消静音。

muteAllRemoteAudio:

muteAllRemoteAudio:
void muteAllRemoteAudio: (BOOL mute)

静音/取消静音所有远端音频

参数	描述
mute	`BOOL`；YES 静音，NO 取消静音。

startLocalAudio

startLocalAudio
void startLocalAudio ()

开始对讲（上行音频）

stopLocalAudio

stopLocalAudio
void stopLocalAudio ()

停止对讲

muteLocalAudio:

muteLocalAudio:

```
void muteLocalAudio:    (BOOL mute)
```

静音本地音频

参数	描述
mute	`BOOL`；YES 静音，NO 取消静音。

startLocalVideo:videoEncoderParams:

startLocalVideo:videoEncoderParams:

```
void
startLocalVideo:videoEncoderP    (TXView* view, TXIoTVideoEncoderParams* videoEnc
arams:
```

开始本地视频采集和推送

参数	描述
view	`TXView*`；用于渲染本地视频的视图。
videoEncoderParams	TXIoTVideoEncoderParams*；视频编码参数。

stopLocalVideo

stopLocalVideo

```
void stopLocalVideo    ()
```

停止本地视频采集和推送

sendPTZCommand:command:speed:

sendPTZCommand:command:speed:

```
void sendPTZCommand:command:speed: (NSInteger channelId, TXIoTPTZCommand command, I
```

发送云台控制指令

参数	描述
channelId	`NSInteger`；通道 ID。
command	TXIoTPTZCommand；云台指令。
speed	`NSInteger`；控制速度。

takeSnapshot:

takeSnapshot:

```
void takeSnapshot: (NSInteger channelId)
```

截图。截图结果通过 `onSnapshotComplete` 回调通知

参数	描述
channelId	`NSInteger`；通道 ID。

startLocalRecording:params:

startLocalRecording:params:

```
void startLocalRecording:params: (NSInteger channelId, TXIoTLocalRecordingParams* pa
```

开始本地录制。录制状态通过 Delegate 回调通知

参数	描述
channelId	`NSInteger`；通道 ID。
params	TXIoTLocalRecordingParams *；录制参数，包含文件路径等。

stopLocalRecording:

stopLocalRecording:

```
void stopLocalRecording:(NSInteger channelId)
```

停止本地录制

参数	描述
channelId	`NSInteger`；通道 ID。

监控事件回调

`TXIoTMonitorSessionDelegate` 协议定义监控事件回调，所有方法均为 `@optional` 。

TXIoTMonitorSessionDelegate

回调	说明
onError:errorCode:errorMessage:	实时监控过程发生错误，错误码见 TXIoTErrorCode 。
onSessionEstablished	实时监控会话建立成功。
onSessionReconnecting	实时监控会话重连中。
onSessionRecovery	实时监控会话恢复。
onRemoteStreamAvailable:available:	远端流可用性变化。
onRenderFirstFrame:	首帧渲染完成。

onPlayStateChanged:state:	播放状态变化。
onSnapshotComplete:image:errorCode:	截图完成。
onLocalRecordBegin:errorCode:storagePath:	本地录制开始。
onLocalRecording:durationMs:storagePath:	本地录制进行中。
onLocalRecordComplete:errorCode:storagePath:	本地录制完成。

监控相关数据结构与枚举

TXIoTVideoEncoderParams

TXIoTVideoEncoderParams

字段	说明
resolution	视频编码分辨率。

TXIoTLocalRecordingParams

TXIoTLocalRecordingParams

字段	说明
filePath	本地录制文件保存路径。

TXIoTStreamType

TXIoTStreamType

枚举	取值	说明

TXIoTStreamTypeHD	0	高清
TXIoTStreamTypeSD	1	标清

TXIoTPTZCommand

TXIoTPTZCommand

枚举	取值	说明
TXIoTPTZCommandUp	0	上
TXIoTPTZCommandDown	1	下
TXIoTPTZCommandLeft	2	左
TXIoTPTZCommandRight	3	右
TXIoTPTZCommandZoomIn	4	放大
TXIoTPTZCommandZoomOut	5	缩小
TXIoTPTZCommandStop	6	停止

TXIoTPlayState

TXIoTPlayState

枚举	取值	说明

TXIoTPlayStatePlaying	0	正在播放
TXIoTPlayStateLoading	1	加载中
TXIoTPlayStateStopped	2	已停止

TXIoTVideoResolution

TXIoTVideoResolution

枚举	取值	说明
TXIoTVideoResolution_640_360	0	横屏 640×360
TXIoTVideoResolution_960_540	1	横屏 960×540
TXIoTVideoResolution_1280_720	2	横屏 1280×720
TXIoTVideoResolution_1920_1080	3	横屏 1920×1080
TXIoTVideoResolution_360_640	4	竖屏 360×640
TXIoTVideoResolution_540_960	5	竖屏 540×960
TXIoTVideoResolution_720_1280	6	竖屏 720×1280
TXIoTVideoResolution_1080_1920	7	竖屏 1080×1920