

- 1 授权库 SDK 集成与接口说明
 - 1.1 免责声明与责任边界
 - 1.2 1. 概述
 - 1.2.1 1.1 文档定位
 - 1.2.2 1.2 关联文档与附表
 - 1.2.3 1.3 参考实现说明
 - 1.3 2. 参考示例与集成入口
 - 1.3.1 2.1 初始化行为要点
 - 1.3.2 2.2 授权请求参数 (示例 `AuthLibRequest`)
 - 1.4 3. `AuthCollection` 类: 事件与授权 API
 - 1.4.1 3.1 事件 (建议全部订阅并转发到宿主日志或 UI)
 - 1.4.2 3.2 方法签名约定
 - 1.4.3 3.3 串口: 工单 / Token 一次授权 (含配件变体)
 - 1.4.4 3.4 TCP: Linux / Linux 4G 一次授权
 - 1.4.5 3.5 串口: 二次检 (对应各协议工单 / Token 实现类)
 - 1.4.6 3.6 TCP: 二次检 (Linux / Linux 4G)
 - 1.5 4. 状态码体系与附表
 - 1.5.1 4.1 两类状态码
 - 1.5.2 4.2 与 `CodeDescription` 的关系
 - 1.5.3 4.3 状态码全量附表 (若有)
 - 1.6 5. 推荐时序
 - 1.7 6. 重试与幂等
 - 1.8 7. 安全与合规
 - 1.9 8. 异常与回调
 - 1.10 9. 兼容性策略
 - 1.11 10. 交付物说明
 - 1.12 11. 文档维护与支持

1 授权库 SDK 集成与接口说明

文档副标题：企业对外版

项	内容
文档编号	DOC-TUYA-AUTHCOLLECTION-INT-001
文档类型	对外集成与接口说明
适用程序集	Tuya.AuthCollection.dll
目标读者	第三方集成商、产测与宿主应用开发人员及技术负责人
版本说明	请以随附程序集 AssemblyVersion 与《发行说明》为准，确保文档与所用二进制一致；对外发放前应 与交付物对齐。

1.1 免责声明与责任边界

- 本文档描述 **Tuya.AuthCollection** 在典型集成场景下的接口与事件约定，**不构成**对具体模组、固件或云端策略的承诺。
- 账号、Token、密钥等凭据须由集成方安全管理；文档中的示例参数均为占位，**禁止**在生产环境直接复用。
- 与依赖程序集（含 **TuyaCloudIfLib** 等）的版本关系以 **交付包清单** 与《发行说明》为准；升级前须在受控环境完成回归。

1.2 1. 概述

1.2.1 1.1 文档定位

本分册说明 **Tuya.AuthCollection**（授权库）SDK 的 **集成方式与公开接口**，面向对外技术交付与验收。

1.2.2 1.2 关联文档与附表

- 《授权库 SDK 升级与接口说明》分册：**Tuya.SdkUpgrade** 所承载的程序集在线升级能力。
- 《涂鸦独立授权库（IndependentAuth）集成指南》分册：**Independentauth.dll**（命名空间 **Tuya.IndependentAuthorize**）。
- 状态码附表**：**CommonMsgCode** / **AuthMsgCode** 全量列表（数值、常量名及说明摘要）随本说明一并提供，版本与随附程序集一致；更新以各版本《发行说明》为准。

1.2.3 1.3 参考实现说明

本分册与 SDK 企业总册在章节结构与表格版式上保持一致。参考实现对应 **Program.cs**、**AuthCollectionExample.cs**（无 **AuthLib** 门面，直接调用 **AuthCollection**）；具体以随附参考源码为准（若有）。

1.3 2. 参考示例与集成入口

参考示例演示「初始化 → 授权请求 → 结果回调 → 日志与错误码映射」的推荐集成方式。示例为控制台程序，便于在无硬件环境下验证参与错误码；生产环境请替换为应用配置与真实设备参数。

能力项	说明
初始化	创建 <code>AuthCollection</code> ，订阅库内事件并接入日志； 本 Demo 见 <code>AuthCollectionExample.SubscribeAllEvents</code> ；运行参数由示例内常量集中维护，生产侧请改为配置下发。（企业总册中的 <code>AuthLib.AuthLib_Init(callbacks)</code> 为历史门面，本 Demo 未引用。）
授权请求	生产侧 应直接调用下文 <code>AuthCollection</code> 上与产品形态匹配的授权 API； 本 Demo 见 <code>AuthCollectionExample.Run</code> 内对 <code>LinuxTokenAuth_TCP</code> 等方法的调用。
回调	通过订阅 <code>AuthProgressEvent</code> 等事件获得进度与结果； 本 Demo 将事件输出至控制台。（企业总册中的 <code>AuthLibCallbacks</code> 为历史门面回调形态。）
状态落盘	单字节 ASCII： <code>1</code> 表示成功， <code>0</code> 表示失败或超时；可由宿主在回调中自行落盘；默认文件名由示例常量指定，可改为业务路径。（本 Demo 源码未演示落盘，集成时按需补充。）
日志与错误码	中文描述优先使用 <code>CodeDescription.GetCodeDesc(code, LanguageType.Chinese)</code> （ 未收录的码返回空字符串 ，可回退附表或 <code>AuthResultModel.MessageDetail</code> ）。

1.3.1 2.1 初始化行为要点

- 在初始化阶段完成 `AuthCollection` 构造与事件订阅，将进度、日志、消息、授权数据等事件转发至宿主 UI 或日志系统。
- 示例中的占位地址、设备标识等**仅用于日志说明**；生产环境须使用受控配置，并避免在日志中输出完整密钥或 Token。
- 生产集成应在读取配置后进行参数校验（超时范围、路径合法性等），再调用库接口。

1.3.2 2.2 授权请求参数（示例 AuthLibRequest）

字段	类型	说明
Kind	枚举	SimulateSuccess / SimulateFailure / SimulateTimeout / BluetoothOrder 等示例分支。
PortName	字符串	串口类授权使用；空串口在示例中用于触发典型入参校验错误码（如未配置串口）。
BaudRate	整数	0 表示使用示例默认波特率（如 9600）。
OrderCode、CloudRequestId、DeviceSn、SftVersion	字符串	可覆盖示例占位；为空时回落到示例内默认占位串。

Demo 对照：本仓库未包含 AuthLib 程序集；Token 等场景下请使用 AuthInfoModel / CloudAuthReqModel（及 TuyaCloudIf.GetTokenInfo 等）构造与上述字段语义等价的入参，详见 AuthCollectionExample.cs。

成功判定（以蓝牙工单为例）：BluetoothAuth 返回 true 且 AuthResultModel.Code == CommonMsgCode.AuthOK 时，应触发成功回调并写入成功状态；否则走失败回调并写入失败状态。**禁止**仅依据 bool 或仅依据 Code 单条件判断。

1.4 3. AuthCollection 类：事件与授权 API

类命名空间：Tuya.AuthCollection。集成方实例化 AuthCollection 后，订阅下列事件并调用与产品形态匹配的授权方法即可。

1.4.1 3.1 事件（建议全部订阅并转发到宿主日志或 UI）

事件	委托类型	说明
AuthProgressEvent	ProgressEventHandler	授权进度、状态。
AuthLogEvent	LogEventHandler	授权过程日志。
AuthMessageEvent	MessageEventHandler	授权状态详细信息。
AuthDataEvent	AuthDataEventHandler	授权数据信息。

调用授权方法时，库会将上述事件与当前授权实现关联，以便统一回调到宿主。

1.4.2 3.2 方法签名约定

- **串口类**: `bool MethodName(string portName, int baudRate, AuthInfoModel authInfo, out AuthResultModel retMsg, out int successAmount)`
- **TCP 类 (Linux / Linux 4G)** : `bool MethodName(string hostName, int port, AuthInfoModel authInfo, out AuthResultModel retMsg, out int successAmount)`

返回值 `bool` : 表示库内流程是否按预期执行完毕; **业务是否授权成功**须结合 `retMsg.Code` 判断 (典型成功为 `CommonMsgCode.AuthOK` 或二次检场景下的 `SecondCheckOK` 等, 见附表)。
successAmount : 与多文件、分包或多从设备写入等相关时的成功计数; 单次串口授权多为 `0` 或 `1`, 以具体授权路径行为为准。

1.4.3 3.3 串口: 工单 / Token 一次授权 (含配件变体)

方法	说明 (与随附程序集文档注释一致)
<code>BluetoothAuth</code>	蓝牙工单授权。
<code>BluetoothTokenAuth</code>	蓝牙 Token 授权。
<code>BluetoothAuth_Product</code>	蓝牙工单授权 (配件)。
<code>BluetoothTokenAuth_Product</code>	蓝牙 Token 授权 (配件)。
<code>WifiAuth</code>	Wi-Fi 工单授权。
<code>WifiTokenAuth</code>	Wi-Fi Token 授权。
<code>ThreadAuth</code>	Thread 工单授权。
<code>ThreadTokenAuth</code>	Thread Token 授权。
<code>PlcAuth</code>	PLC 工单授权。
<code>PlcTokenAuth</code>	PLC Token 授权。
<code>ZigbeeAuth</code>	Zigbee 工单授权。
<code>ZigbeeTokenAuth</code>	Zigbee Token 授权。
<code>LinuxAuth</code>	Linux 工单授权 (串口链路)。
<code>LinuxTokenAuth</code>	Linux Token 授权 (串口链路)。

1.4.4 3.4 TCP: Linux / Linux 4G 一次授权

方法	说明
LinuxAuth_TCP	Linux 工单授权（TCP）。
LinuxTokenAuth_TCP	Linux Token 授权（TCP）。
Linux4GAuth_TCP	Linux 4G 工单授权（TCP）。
Linux4GTokenAuth_TCP	Linux 4G Token 授权（TCP）。

1.4.5 3.5 串口：二次检（对应各协议工单 / Token 实现类）

方法	说明
BlueSecondCheck	蓝牙工单二次检。
BlueTokenSecondCheck	蓝牙 Token 二次检。
WifiSecondCheck	Wi-Fi 工单二次检。
WifiTokenSecondCheck	Wi-Fi Token 二次检。
ThreadSecondCheck	Thread 工单二次检。
ThreadTokenSecondCheck	Thread Token 二次检。
PlcSecondCheck	PLC 工单二次检。
PlcTokenSecondCheck	PLC Token 二次检。
ZigbeeSecondCheck	Zigbee 工单二次检。
ZigbeeTokenSecondCheck	Zigbee Token 二次检。
LinuxSecondCheck	Linux 二次检（串口）。
LinuxTokenSecondCheck	Linux Token 二次检（串口）。

1.4.6 3.6 TCP: 二次检（Linux / Linux 4G）

方法	说明
LinuxSecondCheck_TCP	Linux 工单二次检（TCP）。
LinuxTokenSecondCheck_TCP	Linux Token 二次检（TCP）。
Linux4GSecondCheck_TCP	Linux 4G 工单二次检（TCP）。
Linux4GTokenSecondCheck_TCP	Linux 4G Token 二次检（TCP）。

1.5 4. 状态码体系与附表

1.5.1 4.1 两类状态码

类型	典型数值范围	用途
CommonMsgCode	-1 , 10001 ~ 10045 （当前版本；无 10044）	框架与通用过程：初始化、入参校验、固件交互异常、二次检 框架码、网络等。
AuthMsgCode	20000 ~ 200348 （随版本递增）	产测与业务子步骤： 写证书、HomeKit、 FFS、Matter、 MAC、国家码等细粒 度结果。

AuthResultModel.Code 及事件上抛出的业务码可能为上述任一类常量；集成侧应统一按 int 记录并映射展示文案。

1.5.2 4.2 与 CodeDescription 的关系

CodeDescription.GetCodeDesc(int code, LanguageType type) 维护常用码的中英文字典；字典未覆盖的码不会报错，但返回空字符串。此时请使用：

- 1. 随包 状态码全量附表 （ CommonMsgCode / AuthMsgCode 分册或电子附录）中的说明列；或
- 2. 自行按附表维护扩展字典。

1.5.3 4.3 状态码全量附表（若有）

附表	内容
CommonMsgCode 全量表	CommonMsgCode 全部常量：数值、名称、说明摘要。
AuthMsgCode 全量表	AuthMsgCode 全部常量：数值、名称、说明摘要。

随包若以 印刷件或电子版 提供上述附表，与本说明为同一版本来源；以《发行说明》所载版本为准。

与程序集保持一致：AuthMsgCode 中个别常量取值在源码中可能呈现非连续位模式（例如 ReadRFFail 、 CheckFfsInfoOk 等），以随附程序集实际数值为准；集成时请勿自行改动约定数值。

1.6 5. 推荐时序

```
sequenceDiagram
    participant Host as 宿主
    participant Facade as 集成门面
    participant Lib as AuthCollection
    participant Dev as 设备/网络

    Host->>Facade: 初始化(callbacks)
    Facade->>Lib: 构造并订阅事件
    Host->>Facade: 发起授权请求
    Facade->>Lib: 调用具体授权 API
    Lib->>Dev: 指令与数据交互
    Lib-->>Facade: 进度/日志/消息事件
    Lib-->>Facade: AuthResultModel
    Facade->>Host: 成功/失败回调
    Facade->>Host: 状态落盘
```

Demo 对照： `Program.cs` 中在授权前调用 `UpgradeExample.RunUpgradeCheckAndDownload`；`AuthCollectionExample.Run` 内为 `UserLogin` → `GetTokenInfo`（若需要）→ 订阅事件 → 调用具体授权 API。门面角色由宿主代码承担，图中 **Facade** 可对应 `AuthCollectionExample` / 业务层。

1.7 6. 重试与幂等

- **库内：**授权实现中对单步指令通常具备发送/接收重试与间隔配置（见 `AuthConfigModel` 及程序集内相关配置类型）。
- **集成方：**对串口占用、网络闪断可采用退避重试；对明确业务拒绝应避免高频重试。
- **幂等：**云端侧建议以 `cloudAuthReq.requestId`（或工单号 + SN + 时间窗口）做去重；设备侧重复授权须符合产线工艺。

1.8 7. 安全与合规

- 生产链路须使用 **HTTPS**；升级与清单下载的安全要求见《授权库 SDK 升级与接口说明》。
- **禁止**在源码、配置文件或日志中写入完整密钥、Token 或客户隐私明文。
- 日志中的 URL、SN 等建议截断或脱敏。

1.9 8. 异常与回调

- 库内对部分异常会转换为 `AuthResultModel`；集成方仍应对公开 API 外层增加 `try/catch`，防止未预期异常导致宿主崩溃。
- 回调内避免长时间阻塞与未处理异常；必要时投递到宿主工作线程处理。

1.10 9. 兼容性策略

- 配置与远端清单：**新增键应为可选**；旧版宿主应忽略未知键。
- 协议字段：新增字段须文档化，默认在缺省时不改变既有行为；新增错误码须使用新数值，不得变更既有码语义。

1.11 10. 交付物说明

标准交付包通常包含：`Tuya.AuthCollection.dll`、依赖程序集、参考示例（若有）、本说明（《授权库 SDK 集成与接口说明》）与《授权库 SDK 升级与接口说明》、状态码附表（电子版及/或印刷件）。具体以《发行说明》与交付清单为准。

参考工程（若有）：除上述 SDK 标准交付物外，可提供示例源码及依赖引用说明，版式与本说明一致，仅供集成对照。

1.12 11. 文档维护与支持

- **文档维护**：与交付程序集及《发行说明》同步修订；对外发放前应核对 **文档编号** 与程序集版本一致。
- **技术支持**：请联系贵司对接的 **涂鸦商务/技术支持渠道**。