

**Guidance for Development Process
of Radar Sensor Docked with Tuya Cloud**

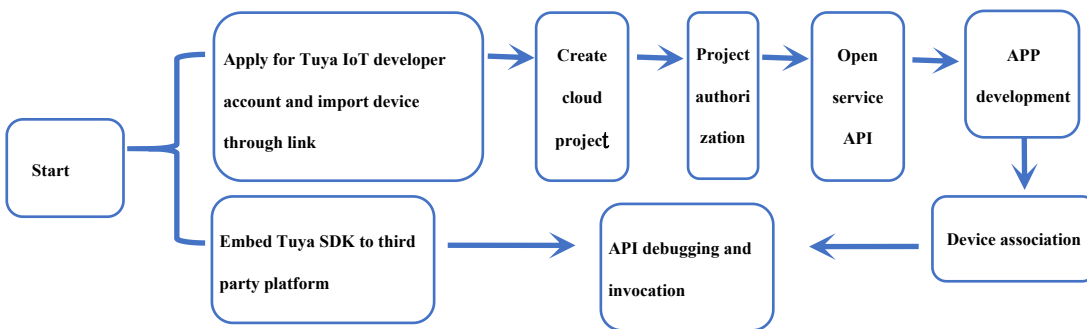
Guidance for development process of radar sensor docked with Tuya Cloud

I. Overview

This document mainly describes the data interaction between third-party service and Tuya platform and before such interaction how the user may import device in Tuya developer background, how to configure related projects of backgrounds and how to transplant relevant SDK to third-party service platform.

Tuya Cloud Platform supports creation of cloud development projects based on custom development and smart house PaaS development.

II. Docking process



III. Import or create device type

A. Create your own developer account at Tuya IoT background and login, open Tuya product link offered by Yufan (Micradar is the English brand name) directly to import. Then, you can see the custom DP related to the product, as shown in the figure below.

Function				
Induction state	Sensitivity	Reset Flag	Motion State	Light Intensity
Detection Data	Check Start	Check End Flag	HW Version	SW Version
Radar ID	Scene	Motion Direction	Motion Value	Nobody Time Set

B. Replicate the device, fill in out its brand name and model, and record PID created.

Copy Product X

* Product Description:

* Product model:

Protocol type: Wi-Fi,Bluetooth LE

Confirm Permission:

- Allow the original developer to continue to help me develop
 - ① The original developer can develop and manage the product
- Allow the original developer to continue to help me purchase
 - ① Original product developers can use product purchase orders to purchase modules and authorization codes
- Allow the original developer to view device logs.
 - ① The original developer can view the device logs of the product for troubleshooting purposes.

IV. Create cloud project

Refer to the link from Tuya Platform as following

<https://developer.tuya.com/en/docs/iot/operation-guide?id=Ka79gg53iivyk>

V. Third-party service platform development

A cloud development project that can help you create Open API or message subscription function in an efficient manner. You may focus on business logic only, but do not need to handle with server side programming or relational database.

The cloud development engineer needs to pull SDK from GitHub repository. The following types of platform are supported respectively:

java - <https://github.com/tuya/tuya-connector>

go - <https://github.com/tuya/tuya-connector-go>

nodejs - <https://github.com/tuya/tuya-connector-nodejs>

<https://github.com/tuya/tuya-iot-python-sdk>

For message queue push, refer to the following link: <https://developer.tuya.com/cn/docs/iot/message-service?id=K95zu0nzdW9cd>.

The following types of platform can be supported separately:

Pulsar Java: <https://github.com/tuya/tuya-pulsar-sdk-java>

Pulsar Go: <https://github.com/tuya/tuya-pulsar-sdk-go>

C# WebSocket:

VI. Updates history

Revision	Release Data	Summary	Author
V1.0_0729	2021/07/29	First draft	Frank