Panel SDK Development Quick Start

Panel Development > Panel SDK Development > 5 Min > Panel SDK Development Quick Start

Version: 20200221

Online Version

Contents

1 Background

> Before getting started, you can briefly understand some basic concepts related to intelligent home

Smart home products generally consist of a main controller MCU, wireless modules (Wifi, ZigBee, Bluetooth Mesh, etc.) and some peripheral circuits (sensors, relays, etc.). Taking the smart socket as an example, the wireless module obtains the instructions issued by the mobile phone App and transmits them to the main controller. The main controller controls the peripheral circuits (such as switch relays) according to the instructions. At the same time, the information monitored by the sensors can also be sent to the wireless module. Cloud until reporting to mobile App for real-time monitoring.

- **Networking module:** Networking module is a module that can be integrated into the target electronic system. Here networking means connecting to the network (Internet, Ethernet, etc.). Networking module is a circuit module that connects the embedded system to the network. . Tuya now has networking modules: WiFi module, GPRS module, and Bluetooth Mesh module. These modules all implement networking functions, but the networking methods are different.
- **MCU (Micro Control Unit):** Micro Controller Unit. The device has its own control board. The function of the product is defined by the control board. The networking module is connected to the control board to realize the networking function.
- **SoC:** System On Chip, there is no MCU in the hardware itself, and the control program is written into the networking module. Generally, this demand is relatively certain and standardized, such as sockets, light bulbs, and so on.
- **Firmware:** Firmware is the program written in EROM (Erasable Read Only Memory) or EEPROM (Electrically Erasable Programmable Read Only Memory). Firmware refers to the device “driver” stored in the device. Through the firmware, the operating system can implement the operation of a specific machine according to the standard device driver. For example, optical drives, recorders, etc. have internal firmware. Firmware is the software that performs the most basic and bottom-level work of a system.
- **DP Point:** The most important thing in product development is mainly for product function. After defining the product function point, that is, the DP point,

we can develop according to the type of DP point.

- Function point: abstraction of specific smart device functions, used to describe product functions and their parameters.
- Function point ID: The function point code. The subsequent device and cloud function data transmission is performed through the function point ID.
- type of data:
  - Bool: A binary variable that is either true or false. Eg: switch function, on / off.
  - Value: suitable for linearly adjustable data. Such as: temperature adjustment, temperature range 20-40 ℃.
  - Enum: custom finite set value. Such as: working gear, low / medium / high.
  - Bitmap (bitmap): It is used for multi-state display. It is generally used for reporting and statistical faults. For specific usage, please refer to FAQs (# FAQs).
  - String: The function points transmitted in the form of strings are generally used for more complicated functions. Only when other types cannot meet the requirements, multiple parties can cooperate and use them.
  - Transparent type (raw): The raw type is simply the string type after base64 encryption. On the control panel side, the process we issue only needs to be consistent with the string. The native side will help us perform base64 encoding and decoding.
- Data transmission type:
  - Can be issued and reported (rw): instruction data can be issued to the device, and device data can be reported to the cloud;
  - Report only (ro): data can only be reported from the device;
  - Delivery only: Data can only be delivered from the cloud;

- **Virtual Device:** Simulated devices can help us to develop quickly within a limited simulation range, but it should be noted that virtual devices **will not actively report DP points and handle the linkage logic**. Take an example of a socket (assuming that it has a master switch and two sub-switch DP points. When we issue a master switch on command, the virtual device will only report

that the master switch is on, and the real device is processed by the firmware Then it will report back the main switch and two sub-switches on).

- **Real Device:** Device with integrated network module and completed firmware logic processing.

In addition, before development, we need you to ensure that Node.js v8 or higher is installed and configured correctly. In addition, we assume that you have development-related experience such as ES2015, React and React-Native.

2 Experience Panel

Use your mobile phone to scan and download Tuya Panel-RN App to register a new user.

2.1 Tuya-Panel-Kit Template

The initial **template project** for Tuya development, the specific code is hosted in Github, you can clone or download View locally.

Scan the following QR code in the `Tuya Panel-RN App`. You can view the currently template panel project.

Figure 1: image.png

2.2 Tuya-Panel-Kit UI Explorer

Tufa panel development ** UI function component browsing library **, the sample code is hosted at Github, you can clone or download View locally.

Scan the following QR code in the `Tuya Panel-RN App`. You can view the currently open functional components of Tuya for use during development and speed up the development of the panel.

3 Create Panel Project

After experiencing the panel, we can start preparing to build our own panel. In actual project development, you will need a series of engineering requirements such as project initialization, development of component libraries, code building, debugging, and packaging. Therefore, we provide a set of development tools to assist the development. Below we take a panel project as an example.

3.1 Install Tuya Evaluate Kit

Tuya provides the official command line tool (CLI), which is deeply integrated with Tuya's panel customization collaboration platform. Please be sure to install it.

```
npm install tuya-panel-kit-cli -g
```

Installing the cli tool through yarn is not supported temporarily

After installation, you can access the `tuya-panel-kit-cli` command from the command line.

You can verify that it was successfully installed by simply running `tuya-panel-kit-cli` and seeing if it displays a copy of the help information for all available commands.

```
$ tuya-panel-kit-cli help
A command line App for developing Tuya panels.

Usage:
  tuya-panel-kit-cli [command]

Available Commands:
  help        Help about any command
  init        init awesome project with name `AwesomeProjectName`
  package     package project

Flags:
  -h, --help   help for tuya-panel-kit-cli

Use "tuya-panel-kit-cli [command] --help" for more information about a
    command.
```

For detailed functions, please refer to Development Tool Documentation

3.2 Initialize template project

In the actual panel project development process, we first need to create a template project through the built-in scaffolding in the Tuya development tool, and then perform panel development based on this template project.

```
tuya-panel-kit-cli init {YourProjectName}

cd {YourProjectName}
yarn
yarn start
```

For the explanation and usage of specific templates, please refer to Template Documents

3.3 Start virtual device

If you want to know what a virtual device is, you can refer to the basic concepts at the top of the document

- How to configure App interface
- Hardware and Embedded Program Development

Before developing a specific product, we need to define a specific DP point and create a product based on it. In addition, we need to add the device to our account before each development. The specific process of creating a product can be reference here。

3.4 Development and debugging

IOS:

1. Open the `Tuya Panel-RN App` to register and log in
2. Click RN debugging in the bottom bar to enter the RN debugging setting page
3. Turn on the slider at the top to enable the RN debugging function, and follow the steps provided by the App to enter **Product ID** / **Local debugging IP address** / **Local debugging port** and **bundle** (For loading local bundles)
4. After inputting, return to the homepage and enter the designated panel to load the local resource pack.

5. If you want to view the debug log or enable hot reload, you can select it by shaking it on the panel interface.

Android:

1. Find the com.ty.panelIdev App that comes with the download and installation
2. Enter the IP address of the development host. After completing the entry, you must **click OK** and turn on the switch.
3. Open `Tuya Panel-RN App` to register and log in
4. After entering the designated panel, the local resource pack will be loaded
5. If you want to view the debug log or enable hot reload, you can select it by shaking it on the panel interface.

If you encounter tips from untrusted enterprise developers, please refer to FAQs. Please also note that you need to start the RN service locally before npm start;

3.5 Using components

The tuya-panel-kit component library dependency has been added inside the scaffolding. If you need to upgrade the version later, you can view the component Update Log select the version to upgrade.

3.6 Bundle

After the project is developed, run the following command to package

```
1 tuya-panel-kit-cli package {YourAwesomeProject}
```

Before the package is checked, the validity of the project is checked once, and the verification will start to build.

When you start packaging, `tuya-panel-kit-cli` will package the selected items into a temporary directory. After successful, the temporary directory will be automatically opened.

Packaging will generate 3 UI packages. Examples of three package names are as follows:

- {name}-android *{rnVersion}*{version}.tar.gz Android panel UI package
- {name}-ios *{rnVersion}*{version}.tar.gz Apple Panel UI package
- {name}-sources.tar.gz

Please submit the obtained UI package to Tuya's panel customization collaboration platform as required.

> If you encounter errors during the packaging process, please fix the errors in the source code of the project before continuing to build, or ask developer@tuya.com

Update Convention

In order to make better use of the Tuya Developer Service, `tuya-panel-kit-cli` has a built-in update check, please try to upgrade to the latest version, so as not to miss more exciting :).

4 Technical Support

If you encounter a problem when using the template panel provided by us that cannot be run or packaged after development, please make sure:

1. `node` version 8 or higher and the required dependencies are installed correctly
2. Install the debug App provided by Tuya
3. Correctly used the `tuya-panel-cli` tool provided by Tuya to initialize and package the template

If you confirm that you still cannot solve the problem after following the documents and steps provided by us, please submit an Issue on our forum or project Github / issues) to help us troubleshoot issues and make it easier for anyone who encounters them later.

5 FAQs

5.1 Uncertified Enterprise Developer

You need to manually trust the enterprise-level applications, specific operations see here

5.2 Uncaught NetworkError: Failed to execute 'importScripts' on 'WorkerGlobalScope'

Please ensure that the real machine and the development computer are on the same network segment, and global VPN is not enabled.

5.3 How to use bitmap

It is generally used to define the fault type to save space. The binary bit indicates whether the corresponding fault exists, and the corresponding decimal is reported.

- Suppose a product has four types of failures:
- No fault: decimal -> 0; binary -> 0000;
- The first fault occurs: decimal -> 1; binary -> 0001;
- The first and second faults occur at the same time: decimal -> 3; binary -> 0011;
- All faults occur simultaneously: decimal -> 15; binary -> 1111;