



# Quick Start of Custom Development

Version: 20240614

## Contents

<b>1</b>	<b>Example</b>	<b>2</b>
<b>2</b>	<b>Become a developer</b>	<b>4</b>
<b>3</b>	<b>Create and set up a cloud project</b>	<b>5</b>
<b>4</b>	<b>Add a device</b>	<b>8</b>
<b>5</b>	<b>Enable message service</b>	<b>9</b>
<b>6</b>	<b>Develop and debug</b>	<b>10</b>
6.1	API calls . . . . .	10
6.2	Message service . . . . .	12

---

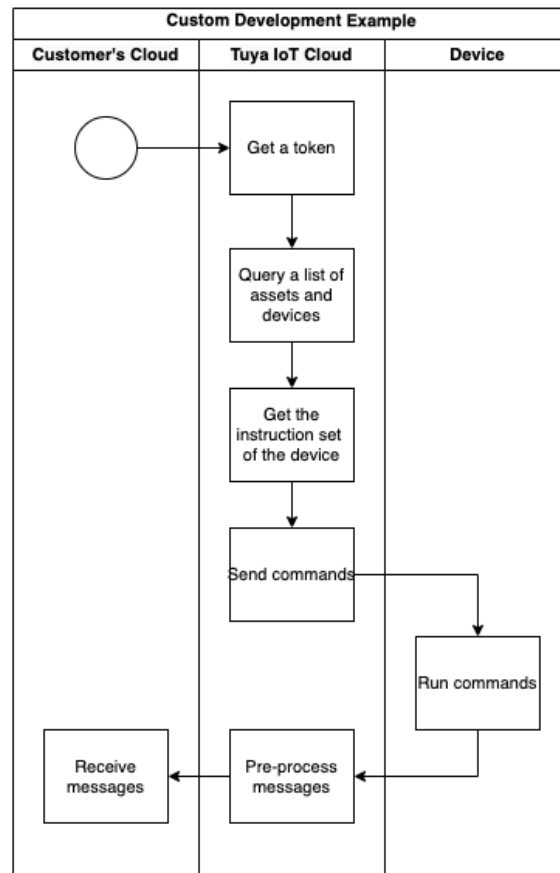
This topic describes how to create and set up a cloud development project in the custom development system, and how to integrate it with Tuya' s cloud development.

## 1 Example

In a cloud project, you can add assets and add devices under the assets. Then, query user and device information and control devices by using OpenAPI and message queues.

1. Become a developer on the [Tuya Developer Platform](#).
2. Create and set up a cloud development project.
3. Add devices.
4. Subscribe to the message service.
5. Develop and debug.
  - a. Get a list of devices under a specified asset.
  - b. Query the instructions supported by a device in the device list.
  - c. Send an instruction to the device.
  - d. With the message service, listen for the messages generated by this instruction.

The following figure shows the interaction logic.



## 2 Become a developer

1. Go to the [account registration page](#) on the Tuya Developer Platform.
2. Follow the guide to register a developer account. The registered account is granted all respective permissions. Keep it properly.

If you are an organization developer, you can follow instructions in [Organization Verification](#) and verify your organization's identity.

### 3 Create and set up a cloud project

The cloud project is a carrier or collection of resources on the Tuya Developer Platform, including devices and users. Resources deployed for each project are isolated from those for other projects.

1. Log in to the [Cloud Development Platform](#).
2. In the left-side navigation bar, choose **Cloud > Development > Create Cloud Project**.



3. Enter the required information as prompted and click **Create**.

To create a custom development project, you must select **Custom** in the **Development Method** field.

Create Cloud Project

X

\* Project Name:

Smart School Project

Description:

Smart School Project

\* Industry:

Education/Campus

\* Development Method:

Custom

\* Data Center ?:

China Data Center X

Cancel

Create

4. On the **Configuration Wizard** page, click **Authorize** and go to quickly create an asset and a user.

Configuration Wizard

---

**Authorize API Services**

The platform recommends some API services, and you can remove and select them as needed. The selected unsubscribed API services will be subscribed to and this project will be granted access the API products.

Select API Services	All	Selected API Service(s) (2)	Delete All
Device OTA Update	>	IoT Core <a href="#">Free Basic Resource Pack</a>	x
IoT Video Live Stream	>	Authorization Token Management	x
Video Cloud Storage	>		
IR Control Hub Open Service	>		
Sleep Band Open Service	>		
Body Fat Scale Open Service	>		
Smart Lock Open Service	>		

Skip Authorize

By default, the APIs that support the specified development method are selected for the project on the platform. You can add or delete the default API services, or select other API services. For more information about API services, see [Manage API Services](#).

5. Enter the required information as prompted and click **Create**. The platform will create an asset and a user accordingly, and authorize the user to access the asset.



#### Configuration Wizard

##### Project Configuration

Creating an asset and user is an indispensable part of the development. Enter the asset and user information. Then, the asset will be created and the user will be automatically granted access to the asset.

\* Data Center:

\* Asset Name:

\* Account Type: ☒ Phone Number ☐ Email

\* Phone Number:

\* Password:

The password must be 6–16 characters in length and contain at least two of the following types: numbers, uppercase letters, lowercase letters, and underscores.

[Skip](#)
[Create](#)

You can also choose to skip this step and create assets and users after the cloud project is created. For more information about assets, see [Manage Assets](#).

- After you create and set up a project, you can view the project in the project list.

Product

App

My Cloud Projects

Upgrade IoT Core Plan >

Create Cloud Project

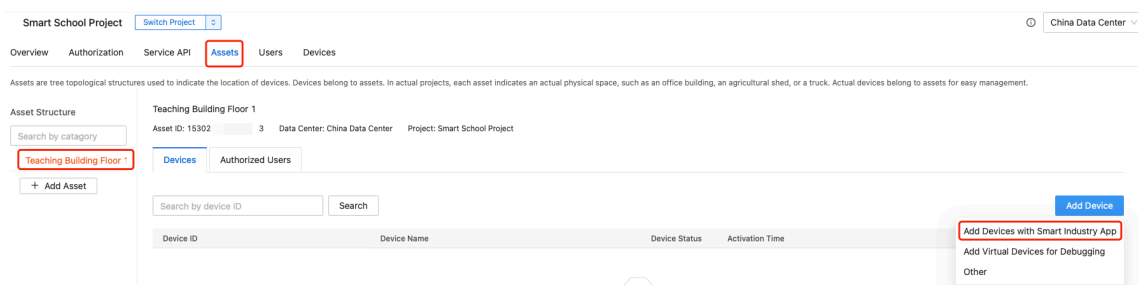
Based on the API products and supporting technical services provided by the Cloud Development Platform, you can quickly build enterprise-level IoT industry solutions by using Powered by Tuya devices. [Learn More](#)

Project Name/Description	Project code	Industry	Creation Time	Operation
Smart School Project	p166243c	Education/Campus	2022-09-08 10:18:32	<a href="#">Edit</a>

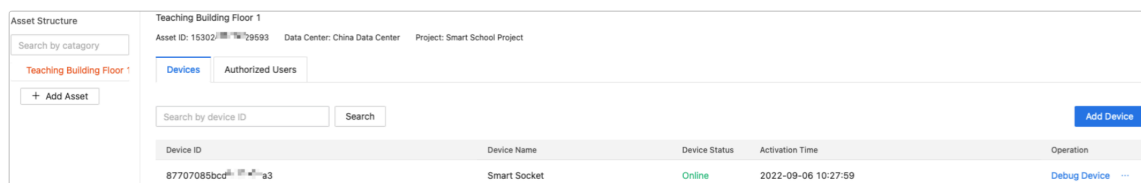
For more information about cloud projects, see [Manage Projects](#).

## 4 Add a device

1. On the page of **Cloud Management**, click a project to enter the details page.
2. Choose **Assets** > **Add Device**, and then select **Add Devices with Smart Industry App**.
3. Scan the QR code to download and install the Smart Industry app, pair and add a device to the asset.



The following figure shows the result of adding a device.



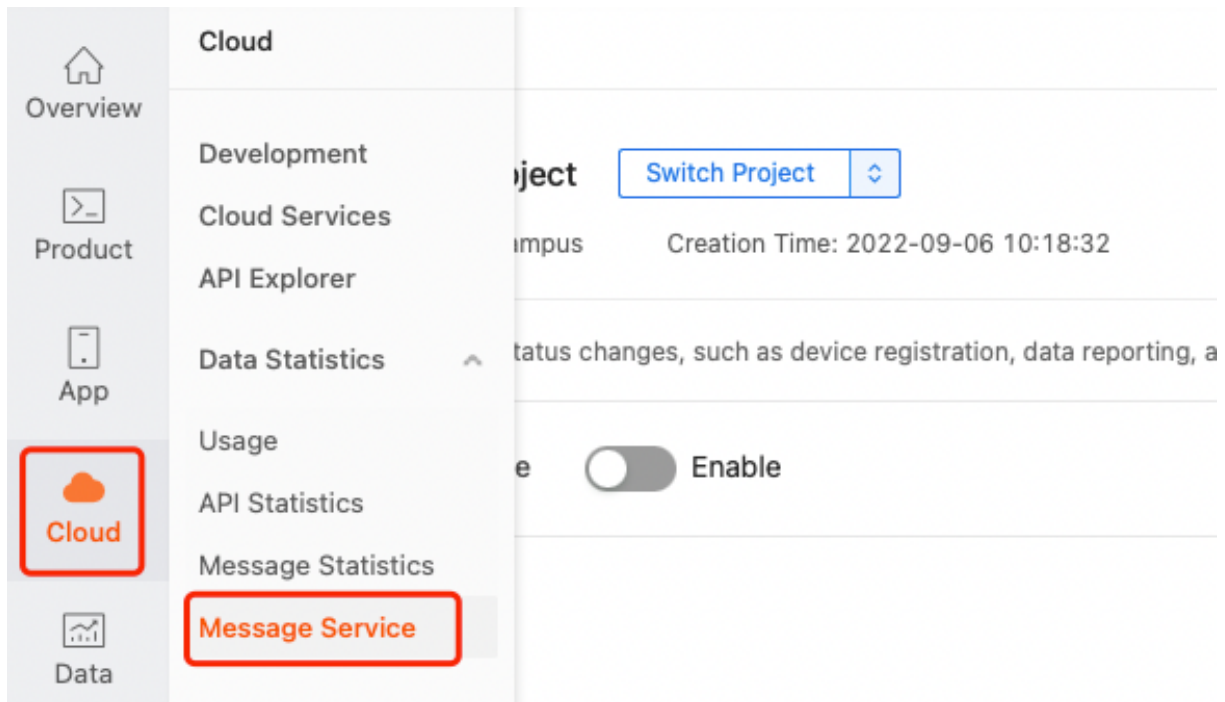
...info

If you do not have a virtual device, you can add a virtual device to try out first. For more information about virtual devices, see [Manage Virtual Devices](#).

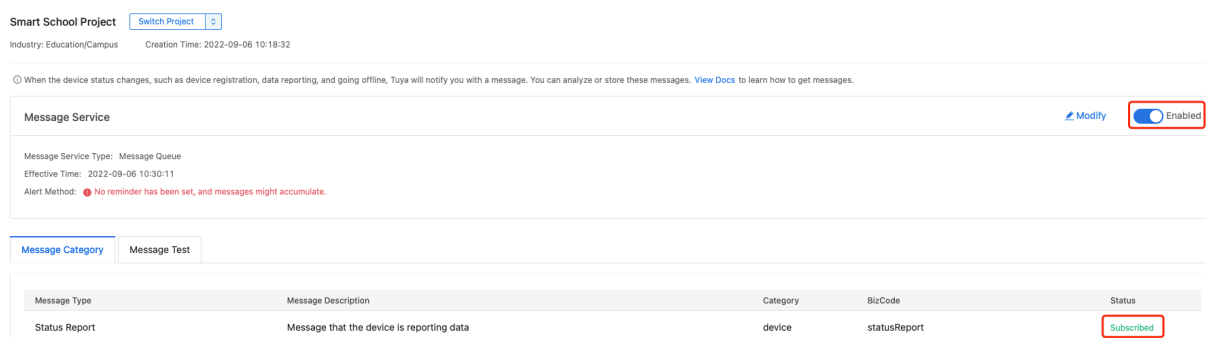
...

## 5 Enable message service

On the page of **Cloud > Message Service**, you can enable this service for the cloud project and configure parameters to get alerts of excessive messages.



The following figure shows the result of enabling the message service.



For more information about the message service, see [Message Service](#).

## 6 Develop and debug

### 6.1 API calls

We will use [API Explorer](#) to complete the above case.

1. Call the [Get Device List](#) API to get a list of devices under the asset.

The screenshot shows the Tuya API Explorer interface. On the left, the 'General Device Management' section is expanded, and the 'Get Device List' API is selected. The 'Params' section shows the following values:

- source\_type: asset
- source\_id: 15302\*\*\*\*\*3 (highlighted with a red box)
- device\_ids: string
- name: string
- category: string
- product\_id: string
- last\_row\_key: string
- page\_size: integer

The 'Debugging Result' tab on the right shows the JSON response. The 'list' array contains one device object, which is highlighted with a red box:

```
{
  "result": {
    "has_more": false,
    "last_row_key": "7D0CCFC92FA0CD9B...",
    "list": [
      {
        "active_time": 1662431279,
        "asset_id": "153024...",
        "category": "cz",
        "category_name": "Wi-Fi Socket",
        "create_time": 1609918144,
        "gateway_id": "",
        "icon": "smart/icon/ay1537005524478pGtw5/158246f...",
        "id": "87707085b*****3",
        "ip": "124.14",
        "lat": "30.8",
        "local_key": "028b596...",
        "lon": "12.8",
        "model": "Wi-Fi Socket F5501-GB",
        "name": "Smart Socket",
        "online": true,
        "product_id": "osdegz...",
        "product_name": "Wi-Fi Socket",
        "sub": false,
        "time_zone": "+08:00",
        "update_time": 1662431282,
        "uuid": "87707085b*****3"
      }
    ]
  },
  "total": 1
},
"success": true,
"t": 1662431533942,
"tid": "1de1b6f52d8c11eda9L..."
}
```

As shown above, there is a socket device with ID 87707085b\*\*\*\*\*3 under the asset.

2. Call the [Get the instruction set of the device](#) API to get the instruction set supported by the device.

Search

General Device Management

- General Devices Control
  - Get the specification...
  - Get the instruction ...
  - Send commands
  - Get Category List
- General Devices management
- General Devices status
- Industrial General Device System
- Smart Home Device System

Get the instruction set of the device

Parameter(Request Method: GET)

Params

- device\_id : 877070851a3

Request URL

```
curl --request GET "https://openapi.tuyacn.com/v1.0/iot-03/devices/877070851a3/functions" --header "signature: 1662431895775" --header "mode: cors" --header "Content-Type: application/json" --header "access_token: 2dc1ad522d400c11edab3f37591e"
```

Response

```
{
  "result": {
    "category": "cz",
    "functions": [
      {
        "code": "switch_1",
        "desc": "Switch 1",
        "name": "Switch 1",
        "type": "Boolean",
        "values": "{}"
      },
      {
        "code": "countdown_1",
        "desc": "The count of Switch 1",
        "name": "The count of Switch 1",
        "type": "Integer",
        "values": "{unit: 's', min: 0, max: 86400, scale: 0, step: 1}"
      }
    ]
  },
  "success": true,
  "t": 1662431895775,
  "tid": "76d88ddf2d8c11edab3f37591e"
}
```

During API debugging, the platform uses the authorization key (Access ID & Access Secret) of the current project to and online resources. Proceed with caution.

The socket supports two functional instructions:

- Switch
- Countdown

The code of the switch instruction is `switch_1`, and the value is of Boolean type. `true`: Turn on. `false`: Turn off. For more information about functional instructions, see [Standard Instruction Set](#).

3. Call the [Send commands](#) API to control the device and turn on the socket.

Search

General Device Management

- General Devices Control
  - Get the specification...
  - Get the instruction ...
  - Get the instruction ...
  - Send commands
  - Get Category List
- General Devices management
- General Devices status
- Industrial General Device System
- Smart Home Device System

Send commands

Parameter(Request Method: POST)

Params

- device\_id : 877070851a3

Body

```
{
  "commands": [
    {
      "code": "switch_1",
      "value": true
    }
  ]
}
```

Request URL

```
curl --request POST "https://openapi.tuyacn.com/v1.0/iot-03/devices/877070851a3/functions" --header "signature: 1662431895775" --header "mode: cors" --header "Content-Type: application/json" --header "access_token: 2dc1ad522d400c11edab3f37591e"
```

Response

```
{
  "result": true,
  "success": true,
  "t": 1662431896046,
  "tid": "f5ab36492d8c11edab3f37591e"
}
```

During API debugging, the platform uses the authorization key (Access ID & Access Secret) of the current project to and online resources. Proceed with caution.

The output window shows that the instruction has been sent to the device.

## 6.2 Message service

1. On the **Message Test** tab, add the device to the message test channel and enable the **Test Channel**.

Smart School Project [Switch Project](#)

Industry: Education/Campus Creation Time: 2022-09-06 10:18:32

ⓘ When the device status changes, such as device registration, data reporting, and going offline, Tuya will notify you with a message. You can analyze or store these messages. [View Docs](#) to learn how to get messages.

Message Service

Message Service Type: Message Queue

Effective Time: 2022-09-06 10:30:11

Alert Method: ● No reminder has been set, and messages might accumulate.

Message Category [Message Test](#)

ⓘ The test channel will only push the message of the test devices. If you need to test, please add your test device to the following list. After a device is added as a test device, the device messages are not pushed to the production environment.

87707 [Add Test Device](#)

Test Channel: ☒

2. Call the [Send commands](#) API again and observe the test channel.

```
{
  "devId": "87707085bccc...fa3",
  "productKey": "osdegzy...ts0",
  "status": [
    {
      "i": "false",
      "code": "switch_1",
      "t": 1660205104435,
      "value": false
    }
  ],
  "protocol": 4,
  "pv": "2.0",
  "sign": "b58a0f1c36b8bda15882c7...03aa"
}
```

As you can see, the listener has returned the message that the socket was turned off.

The message test channel is for trial and debugging only. If you add your device to the test channel, you will not receive messages in the production environment.

Now, you have completed the quick start and can dive deep to the custom development system.