FAQ

Open API > FAQ

Version: 20200328

Online Version

Contents

Q: What are the ways to connect Tuya Cloud?

Tuya: Tuya Cloud currently supports two modes for developers to integrate Tuya Cloud.

1. Simple mode:
    - Tuya Cloud provides a cloud-based docking method based on the oauth2 protocol. Developers can apply for a cloud API on the tuya iot platform, call tuya openapi according to the tuya openapi interface specification, obtain the developer's own user and device data, and control the device through permissions.
    - For this case, the developer needs to create an oem application or create a product on the platform. After that, you can get user data or device data based on your development account.
2. Authorization code mode:

```
- After applying the cloud API key on Tuya cloud platform, developers can call Tuya open interface based on Tuya openapi interface specification to obtain resource data under authorization and resource data within the authorization scope;
- For cloud integration scenarios that do not have OEM applications in the tuya platform, tuya cloud provides an authorization method in the form of an authorization code for developers to obtain tokens. After the user authorizes, the manufacturer can share the relevant permissions of the tuya smart user or smart life user.
```

Q: What is the difference between user rights and device operation rights?

Tuya:

1. User permissions are divided into two categories, one is: the developer manages (adds, deletes, changes, and checks) the user data under his app, and the other is the permissions that the developer grants to the C-end user authorization Carry out operation
2. Device operation permissions: Device operation permissions are currently divided into three categories, one is the developer's operation permission, the other is the family administrator's operation permission, and the last one is the ordinary family's share. Operation authority Among them, the permissions for developer operations can be refined into three categories:

```
- Product Dimension Permission: The product to which the device belongs belongs to the developer. Under the Tuya Cloud IoT account, the developer has operation permissions for this type of device.
- application Dimension Permissions: The users bound to the device belong to the developers' users in Tuya Cloud application, and the developers have the operation permission to the messages of such devices.
- Authorized dimension permissions: The user grants the corresponding device resource permissions to third-party developers, and the developers indirectly obtain the operation permissions of these devices.
```

Q: What is the app schema and where can I get it?

Tuya: The app schema is the unique identifier of the app application. It is used to associate the uniqueness of the application. Users who use the cloud API to register users will need this identifier for query users. The acquisition method is different for different scenarios:

- If you are an OEM app customer, in the App workbench of the IoT platform, the package name is obtained by default. The package name is com.xx.yy, and the schema is xxyy. Named: iotesmart;
- If you are an APP SDK customer, in the App workbench of the IoT platform, find and create a good SDK or create an SDK, and the channel ID is the schema.

Q: When calling the interface, an error occurs: “2008: command or value not support” How to solve it?

Tuya: Please provide the device id, Tuya’s technical staff will evaluate the feasibility of the instruction standardization of the product and configure the standardization in time.

Q: Does Tuya open MQTT service?

Tuya: Cloud-to-cloud docking currently does not open MQTT, device control uses HTTP API, real-time message reception supports message queues: Pulsar;

Q: How to prevent deauthentication attacks on nodes?

Tuya: In order to prevent malicious tampering of the API call process, calling any API needs to carry a signature. Tuya Cloud according to the request parameters, signature verification, illegal signature request will be rejected.

Q: How many requests can a user send to the server per second?

Tuya: Tuya Cloud provides the interface current limiting dimension, as shown below:

1. application layer traffic: 500,000 times / day
2. Interface-level traffic: 500 times / second

Detailed description:

1. application traffic is the number of times the calling key assigned to the developer can call the interface each day. Once the interface request is started, as long as the request reaches the Tuya server, the interface request will be counted once regardless of success or failure. In order to prevent a single user from making unlimited requests to the top server, each application limits the total number of requests;
2. Interface traffic is an interface for each developer to call the API itself, and traffic protection will also be set, such as obtaining device information APIs to prevent excessive calls and cause back-end failures. This limit for API traffic is a system protection limit.

Q: Is node IpV6 enabled and will it provide low-power network support?

Tuya: Tuya Cloud ipv6 protocol does not require specific products to support it. Specific equipment is required to support low-power networks, such as zigbee.

Q: After the network is interrupted / reset, what method is used to resynchronize the device, what is the delay in this process, and does it scale with the network scale?

Tuya: The network is interrupted. After the network is restored, the device will automatically connect. The process is on the second level. In addition, tuya cloud also has the ability to dynamically expand based on network load.

Q: On the IoT.tuya.com website, even if 5 devices have been bound for a given user, after logging in we find that the number of activated devices is 0. Why does it appear to be 0?

Tuya: You need to confirm which region you are currently in. Device activation data varies from one data center to another. And you also need to confirm that the device-related pid is under your IoT account. If you do n't have it, you ca n't see the activation data in your account. Only the product manufacturer owns this device activation data.

Q: We use the Pulsar Client in the GO language SDK to listen to messages from the device. We tried to open and close, but we haven't received any messages yet. Can you verify after we provide these keys?

Tuya: Please provide the device ID. We will check it.

Q: What data types can message push support

Tuya: Common data types: device online, device offline, device name modification, device function point name modification, device binding user, device deletion, device event notification. Two background configuration data types: upgrade event, user event (register, remove).

Q: Validity period of message push data?

Tuya: If no consumption is exceeded for more than 2 hours, it will be discarded.

Q: Can I cancel the authorization after the authorization code mode access authorization?

Tuya: The interface for deauthorization is not yet open. The current policy is: the authorization expires after 2 hours. In the future, the interface for canceling expired tokens and the authorization management in the Tuya official app can be opened.

Q: How to get the value of user uid?

Tuya: Most customers are taking the uid parameter value and will use the uid returned by the token. This value is actually of no practical significance to the user and needs to be ignored. The real value should be obtained by obtaining the user list under the application and then checking the corresponding User's uid.

Q: After the cloud user has registered, the app can't log in, how can I solve it?

Tuya: This problem occurs. The most likely reason is that the password field is entered in plain text when registering, and the api requires MD5 encrypted cipher text. When you solve it, you need to register the user again. The password will be replaced.

Q: Validity period of token or sign?

Tuya: The token is valid for 2 hours and the sign is valid for 5min.

Q: Do you need to find the Tuya mode for mqtt message push?

Tuya: Please provide the client_id of the iot developer account, the opening region (China, Europe, the United States), the opening dimension (product, application), and the push method (select mqtt).

Q: What is the subscription type of Yunyun docking Pulsar, exclusive subscription, shared subscription, invalid backup subscription?

Tuya:

1. Exclusive subscription-there can only be one consumer at a time.
2. Shared subscription-can be subscribed by multiple consumers, each consumer receives a part of the message.
3. failover subscription-allows multiple consumers to connect to the same topic, but only one consumer can receive messages. Only when the current consumer fails, other consumers start receiving messages.

The Tuya puslar belongs to the third type, and the invalid backup subscription is invalid.

Q: The calling interface has an error code: 1106. How to resolve the permission deny?

Tuya: There are several situations

1. The device to be controlled is not an OEM App created by the developer on the Tuya Open Platform; or the Tuya Smart APP is used to pair the device and then control the device through an interface call.

   solution:

   ```
   1. First check if the device is paired with Tuya Smart APP, and
      then use the device Id in Tuya Smart APP for control;
   2. Then check whether the Id of the device to be controlled is
      correct and whether it is from an OEM APP created by its own
      developer account or an APP developed based on the SDK;
   3. In the end, you need to make sure that the device and product
      created by the OEM APP created under your developer account can
       be controlled through the Yunyun docking API call.
   ```

2. The device requested to be controlled has been removed, restored to factory settings, or re-configured to the network.

   solution:

   ```
   1. If the device has been removed, please re-configure the network
       and add it in the APP;
   2. Re-obtain the device list and refresh the device ids to be
      controlled.
   ```

3. Interface parameter passing error.

   solution:

```
Check if the following parameters are correct:
1. The device id is wrong. Note that the device id is not uuid.
2. The area url is wrong. Pay attention to the area where the
   device is paired. At the same time, call the interface address
   of the corresponding area.
```