Contents [hide]

tuya 20250618 Space Cloud Services API Reference

Specifications

- Version: 20250618
- Online Version

Product Information

This product allows users to query the list of devices in spaces based on space IDs, product IDs, and product categories. Users can specify whether to recurse into subspaces for detailed device information. A space defines the geographical location, area, and layout information of IoT scenes, and displays the topological relationship between various smart devices in a tree structure. You can add physical resources (mainly smart devices) to each node of the tree structure. For example, lights, air conditioners, and video security devices in rooms.

Functionalities

Flexibly scalable and universal space models across business semantics can cope with business forms of all industries and scenarios. Thus, you can achieve more efficient solutions in diversified scenarios. A space plays multiple roles in IoT application development, including but not limited to:

- **Map a physical space**: For example, define a building.
- **Manage user permissions**: After defining a space, you can respectively link the space with users and resources, and thus bridge the relationship between resources and users based on the space.
- **Manage physical resources**: Under the space, you can add various types of resources, such as devices and mesh networks.

Structural features

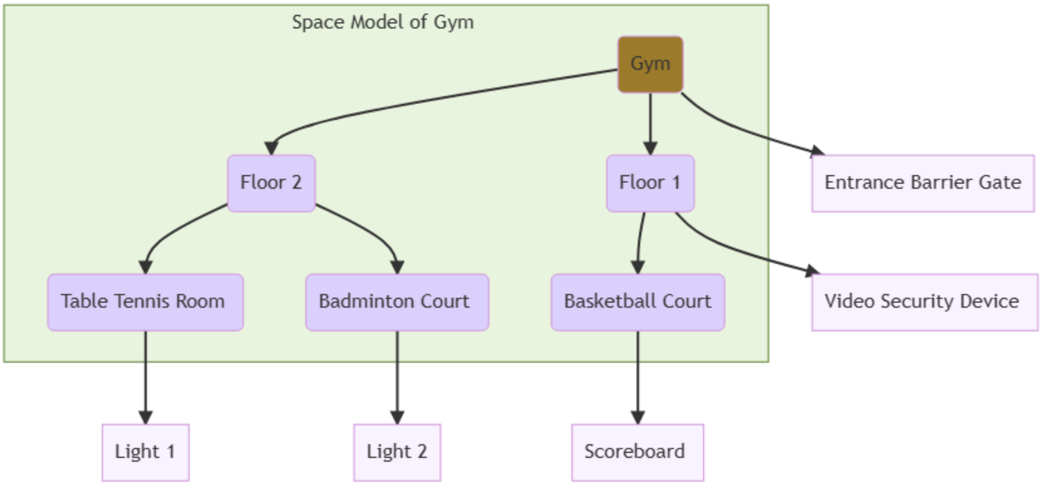
The hierarchical structure constructed by the **space** has the following features:

- A tree-like multi-level structure is presented.
- Each space has only one root node.
- Each node has only one parent node and can have multiple child nodes.
- Resources can be added to each level of nodes.
- Each resource can be added under multiple nodes.
- Under the same space tree, a gateway and its sub-devices can enter different levels of space nodes.

Examples of application

Example 1: Space layout and device linkage of a gym

As a multifunction venue, a gym contains different levels of spaces like the gym hall, floors, and rooms for ball games. You can create the desired spaces in the space model, helping implement device linkage. For example, control lights and a basketball court scoreboard. The following figure shows an example of a space structure. The gym is the root node, and the first floor (including the basketball court) and the second floor (including the table tennis room and badminton court) are child nodes.

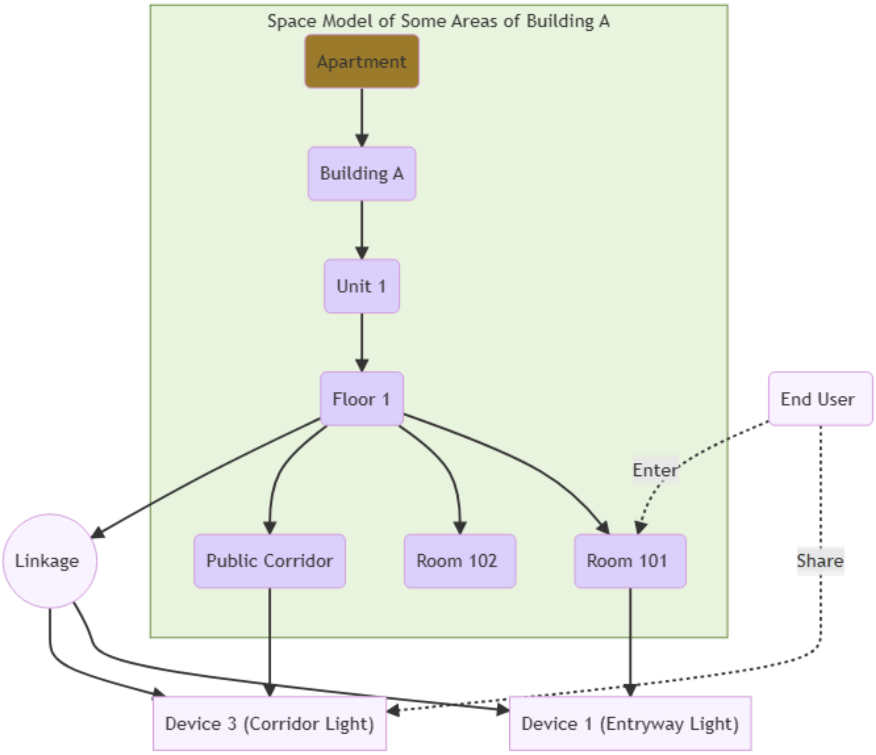


By creating the structure on top of the space model, you can:

- Set the gym as the topmost layer of the entire space, which consists of two floors.
- Specify the basketball court and its scoreboard on the first floor, as well as video security devices.
- Specify the table tennis room and badminton court on the second floor, as well as the corresponding lights.
- Utilize the barrier gate as an entry to access control of the gym, facilitating scene linkage of lights and the scoreboard.

Example 2: Digital space construction of smart apartment

As a popular business scenario, smart apartment can involve spaces in various dimensions, such as apartments, units, buildings, room numbers, and public areas. A resident needs permission to devices not only in his room, but also in the public space area. This way, implement the linkage between devices in the specified room and devices in the public area.The following figure shows an example of a space structure. The apartment is the root node, and Building A, Unit 1, and Floor 1 (including rooms 101 and 102) are child nodes.


Space Model of Some Areas of Building A
Apartment
Building A
Unit 1
Floor 1
End User
Enter
Linkage
Public Corridor
Room 102
Room 101
Share
Device 3 (Corridor Light)
Device 1 (Entryway Light)

By creating the structure on top of the space model, you can:

- Set up the five-level structure of apartment > building > unit > floor > room to represent the layout and logical relationship of the physical world.
- Grant users permission to a specific room, such as Room 101.
- Grant users permission to a specific device, such as Device 3.
- Create desired device linkage rules in different spaces.

On top of this space model, you can further extend and iterate on features.

For example

- **Access control systems**: Add an access control system at the entrance of each building so that only authorized personnel can enter the building.
- **Face recognition**: Upgrade conventional door locks to smart locks with face recognition, so residents can enter their rooms more safely and conveniently.
- **Smart home control**: Connect smart home devices (such as lights and air conditioners) with the space model, so residents can control various devices using a specific mobile app or voice commands.
- **Energy management**: Install an energy management system to monitor and control

the energy usage in the rooms, helping save energy and reduce costs.

- **Video security**: Install smart security cameras in buildings and public areas to make the entire smart apartment a safer place.
- **Social interaction**: Provide a social platform for residents to exchange and share information with each other. Thus, the whole community becomes more interactive and friendly.

After more and more extended functionalities are in place, smart apartments provide a more convenient and comfortable living experience, while improving the safety and quality of life for residents.

API list

API name	API endpoint
Delete Space	DELETE:/v2.0/cloud/space/{space_id}
Query Resources in Space	GET:/v2.0/cloud/space/{space_id}/resource
Create Space	POST:/v2.0/cloud/space/creation
Query Space	GET:/v2.0/cloud/space/{space_id}
Query Space List	GET:/v2.0/cloud/space/child
Modify Space Information	PUT:/v2.0/cloud/space/{space_id}
Query Parent-Child Relationship	GET:/v2.0/cloud/space/relation

Limits

Item	Limits
The number of devices under a single space node	100
The number of child nodes in a single space	100
The number of levels in a single space	10 levels

Frequently Asked Questions

What is the purpose of the is_recursion parameter?

The is_recursion parameter specifies whether to recurse into subspaces when querying device details. Setting it to true will include subspaces in the query results, while setting it to empty will only query devices in the current space without recursion.

What are the valid values for the is_recursion parameter?

The valid values for the is_recursion parameter are true (for recursing into subspaces) and empty (for querying only devices in the current space without recursion).

Documents / Resources

 Query Devices in Space Version: 20250618 Online Version	tuya 20250618 Space Cloud Services API Reference [pdf] User Guide 20250618 Space Cloud Services API Reference, 20250618, Space Cloud Services API Reference, Services API Reference, API Reference

References

- User Manual

Tuya

20250618, 20250618 Space Cloud Services API Reference, API Reference, Services API Reference, Space Cloud Services API Reference, tuya

Leave a comment

Your email address will not be published. Required fields are marked *

Comment *

Name

Email

Website

□ Save my name, email, and website in this browser for the next time I comment.

Post Comment

Search:

e.g. whirlpool wrf535swhz

Search