2023

WIFI Tag System API Interface

VER 1.0.5

DALIAN SERTAG TECHNOLOGY CO., LTD

Contents

1. Parameter configuration

1.1. WIFI Parameter configuration

No.	Name	Describe	Default value
1	SSID	AP SSID	
2	password	AP password	

1.2. MQTT Parameter configuration

No.	Name	Describe	Default value
1	Broker IP	MQTT Broker IP address	
2	PORT	MQTT Broker Port	8883
3	username	MQTT Login username	
4	password	MQTT Login password	
5	client id	MQTT client ID	

2. Device data management

No.	Name	Describe	Type	Default value
1	MAC	Device MAC address	String	
2	IP	Device IP address	String	
3	Voltage	Device battery level	Integer	
4	Station.RSSI	Device signal strength	Integer	
5	Station.SSID	AP SSID	String	
6	Station.Password	AP password	String	
7	Mqtt.Broker	MQTT Broker IP address	String	
8	Mqtt.PORT	MQTT Broker Port	Integer	8883
9	Mqtt.Username	MQTT Login username	String	
10	Mqtt.Password	MQTT Login password	String	
11	Mqtt.Client ID	MQTT Client ID	String	
12	Device Type	Device Type	Object	
13	Screen Type	Screen Type	Object	
14	SN	Serial Number	String	
15	SW	Software version number	Integer	
16	HW	Hardware version number	Integer	
17	Sms.display.success	Number of successful statistics	Integer	

		updates		
18	Sms.display.fail	Statistics update failed times	Integer	
19	Sms.network.conn	Count the number of successful connections	Integer	
20	Sms.network.conn_fail	Count the number of failed connections	Integer	
21	Sms.network.discon	Count the number of disconnections	Integer	
22	Sms.reboot	Count system restart times	Integer	
23	DevID	Device ID number	String	
24	UsbState	USB status	Integer	
25	Product	Products	Object	
26	Algorithm	Image algorithm	Object	

3. MQTT interaction protocol between client and server

3.1. Overview of client MQTT protocol

No.	Topic	Describe	Reference section
1	/client/${ApiKey}/action/online	Subscribe to device online messages	
2	/client/${ApiKey}/action/offline	Subscribe to device offline messages	
3	/client/${ApiKey}/action/usb_state	Subscribe to device USB status messages	
4	/client/${ApiKey}/action/button	Subscribe to device key messages	
5	/client/${ApiKey}/action/display	Post device update screen message	
6	/client/${ApiKey}/action/display_reply	Subscribe to device update screen results messages	
7	/client/${ApiKey}/action/battery	Post Get device battery voltage message	
8	/client/${ApiKey}/action/battery_reply	Subscribe to device battery voltage results messages	
9	/client/${ApiKey}/action/led	Post update LED status message	
10	/client/${ApiKey}/action/led_reply	Subscribe to update LED status result messages	
11	/client/${ApiKey}/action/reboot	Post restart news	

12	/client/${ApiKey}/action/reboot_reply	Subscribe to restart result messages	

Note: MQTT message content is encoded in base64

The message received is similar to the following

ewoJIm1hYyI6ICJENDozRDozOToxNzoyQTo4NCIsCgkibXNnSWQiOiAiMTY1MTIxMTE4NTQ2MiIKfQ==

3.2. Client subscribes to device online messages

The TOPIC format of the client's subscription device online message is:

```
/client/${ApiKey}/action/online
```

For example, the user ApiKey is 61a3bd3d4c10ad03f1b15a99, and the subscription TOPIC format is as follows

```
/client/61a3bd3d4c10ad03f1b15a99/action/online
```

The content format of Message information is as follows

```
{
    "mac": "D4:3D:39:17:2A:84",
    "msgId":"1651211185462"
}
```

Message field description

No.	Field	Value Type	Value	describe
1	mac	String		Device MAC address
2	msgId	String	Timestamp	Mark unique

3.3. Client subscribes to device offline messages

The TOPIC format of the client's subscription device offline message is:

```
/client/${ApiKey}/action/offline
```

For example, the user ApiKey is 61a3bd3d4c10ad03f1b15a99, and the subscription TOPIC format is as follows

```
/client/61a3bd3d4c10ad03f1b15a99/action/offline
```

The content format of Message information is as follows

```
{
    "mac": "D4:3D:39:17:2A:84",
    "msgId":"1651211185462"
}
```

Message field description

No.	Field	Value Type	Value	describe
1	mac	String		Device MAC address

2	msgId	String	Timestamp	Mark unique

3.4. The client subscribes to the device USB status feedback message

The TOPIC format of the client's subscription device USB status message is:

```
/client/${ApiKey}/action/usb_state
```

For example, the user ApiKey is 61a3bd3d4c10ad03f1b15a99, and the subscription TOPIC format is as follows

```
/client/61a3bd3d4c10ad03f1b15a99/action/usb_state
```

The content format of Message information is as follows

```
{
    "mac": "D4:3D:39:17:2A:84",
    "msgId":"1651211185462",
    "state": 1
}
```

Message field description

No.	Field	Value Type	Value	describe
1	mac	String		Device MAC address
2	msgId	String	Timestamp	Mark unique
3	state	Integer	1/0	Plug/unplug

3.5. Client subscribes to device key feedback messages

The TOPIC format of the client's subscription device key message is:

```
/client/${ApiKey}/action/button
```

For example, the user ApiKey is 61a3bd3d4c10ad03f1b15a99, and the subscription TOPIC format is as follows

```
/client/61a3bd3d4c10ad03f1b15a99/action/button
```

The content format of Message information is as follows

```
{
    "mac": "D4:3D:39:17:2A:84",
    "msgId":"1651211185462",
}
```

Message field description

No.	Field	Value Type	Value	Describe
1	mac	String		Device MAC address
2	msgId	String	Timestamp	Mark unique

3.6. Client publishes update device screen message

The TOPIC format of the client's updated device screen message is:

```
/client/${ApiKey}/action/display
```

For example, the user ApiKey is 61a3bd3d4c10ad03f1b15a99, and the subscription TOPIC format is as follows

```
/client/61a3bd3d4c10ad03f1b15a99/action/display
```

The content format of Message information is as follows

```
{
    "method": "display",
    "msgId": "1651211643524",
    "apiKey": "61a3bd3d4c10ad03f1b15a99",
    "version": 1,
    "message": {
        "mac": "D4:3D:39:17:42:F6",
        "imgsrc": "https://t7.baidu.com/it/u=1575628574,1150213623&fm=193&f=GIF"
    }
}
```

Message field description

No.	Field	Value Type	Value	Options	Describe
1	method	String	display	required	method
2	msgId	String	Timestamp	required	mark unique
4	version	Integer	1	required	version number
5	message.mac	String	Mac	required	Device MAC address
6	message.imgsrc	String		Optional	1. If not, the system default picture 2. Convert the image to base64 format 3. Image download address
7	message.template	Object		Optional	Choose one of imgsrc and template
8	apiKey	String		required	ApiKey

Note: You can only choose one of the imgsrc and template fields.

Update the screen with a template

```
{
    "method": "display",
    "apiKey": "61a3bd3d4c10ad03f1b15a99",
```

```
        "msgId": "1651211643524",
        "version": 1,
        "message": {
            "mac": "D4:3D:39:17:42:F6",
            "template": {
                "tid": "6347e877ac6dda127fb5fd11",
                "tname": "会议模板",
                "data": {
                    "标题": "A2107",
                    "网址": "www.xxxx.cn",
                    "内容": "成小明：9:00-12:00"
                }
            }
        }
    }
```

Note: The field template is the json object of the template export interface

3.7. The client subscribes to update the device screen feedback result message

The client subscribes to the device screen update result feedback message in TOPIC format:

```
/client/${ApiKey}/action/display_reply
```

For example, the user ApiKey is 61a3bd3d4c10ad03f1b15a99, and the subscription TOPIC format is as follows

```
/client/61a3bd3d4c10ad03f1b15a99/action/display_reply
```

The content format of Message information is as follows

```
{
    "mac": "D4:3D:39:17:42:F6",
    "msgId": "1651211643524",
    "result": 200
}
```

Message field description

No.	Field	Value Type	Value	Describe
1	mac	String		Device MAC address
2	msgId	String	Timestamp	Mark unique
3	result	Integer	200/400/401	Success/Failure/Other

3.8. The client publishes a message to obtain the device battery voltage

The TOPIC format of the message TOPIC issued by the client to obtain the device battery voltage is:

```
/client/${ApiKey}/action/battery
```

For example, the user ApiKey is 61a3bd3d4c10ad03f1b15a99, and the subscription TOPIC format is as follows

```
/client/61a3bd3d4c10ad03f1b15a99/action/battery
```

The content format of Message information is as follows

```
{
    "method": "battery",
    "msgId": "1651211849119",
    "apiKey": "61a3bd3d4c10ad03f1b15a99",
    "version": 1,
    "message": {
        "mac": "D4:3D:39:17:42:F6"
    }
}
```

Message field description

No.	Field	Value Type	Value	Options	Describe
1	method	String	battery	required	method
2	msgId	String	Timestamp	required	mark unique
4	version	Integer	1	required	version number
5	message.mac	String	Mac	required	Device MAC address
6	apiKey	String		required	ApiKey

3.9. The client subscribes to obtain the device battery voltage feedback result message

The TOPIC format of the client's subscription to obtain the device battery voltage result message is:

```
/client/${ApiKey}/action/battery_reply
```

For example, the user ApiKey is 61a3bd3d4c10ad03f1b15a99, and the subscription TOPIC format is as follows

```
/client/61a3bd3d4c10ad03f1b15a99/action/battery_reply
```

The content format of Message information is as follows

```
{
    "mac": "D4:3D:39:17:42:F6",
```

```
        "msgId": "1651211849119",
        "voltage": 410
    }
```

Message field description

No.	Field	Value Type	Value	Describe
1	mac	String		Device MAC address
2	msgId	String	Timestamp	Mark unique
3	voltage	Integer		100 times the battery voltage

3.10. The client publishes updated device LED status messages

The client publishes an updated LED status message in the TOPIC format:

```
/client/${ApiKey}/action/led
```

For example, the user ApiKey is 61a3bd3d4c10ad03f1b15a99, and the subscription TOPIC format is as follows

```
/client/61a3bd3d4c10ad03f1b15a99/action/led
```

The content format of Message information is as follows

```
{
    "method": "led",
    "msgId": "1651211844116",
    "apiKey": "61a3bd3d4c10ad03f1b15a99",
    "version": 1,
    "message": {
        "mac": "D4:3D:39:17:42:F6",
        "red": 255,
        "green": 0,
        "blue": 255,
    }
}
```

Message field description

No.	Field	Value Type	Value	Options	Describe
1	method	String	led	required	method
2	msgId	String	Timestamp	required	mark unique
4	version	Integer	1	required	version number
5	message.mac	String	Mac	required	Device MAC address
6	message.red	Integer	0/255	Optional	Red light off/on
7	message.green	Integer	0/255	Optional	Green light off/on
8	message.blue	Integer	0/255	Optional	Blue light off/on
9	apiKey	String		required	ApiKey

3.11. The client subscribes to update the device LED status feedback result message

The TOPIC format of the client's subscription update LED status result message is:

```
/client/${ApiKey}/action/led_reply
```

For example, the user ApiKey is 61a3bd3d4c10ad03f1b15a99, and the subscription TOPIC format is as follows

```
/client/61a3bd3d4c10ad03f1b15a99/action/led_reply
```

The content format of Message information is as follows

```
{
    "mac": "D4:3D:39:17:42:F6",
    "msgId": "1651211844116",
    "result": 200
}
```

Message field description

No.	Field	Value Type	Value	Describe
1	mac	String		Device MAC address
2	msgId	String	Timestamp	Mark unique
3	result	Integer	200/400	success failure

3.12. Client publishes device restart message

The TOPIC format of the device restart message issued by the client is:

```
/client/${ApiKey}/action/reboot
```

For example, the user ApiKey is 61a3bd3d4c10ad03f1b15a99, and the subscription TOPIC format is as follows

```
/client/61a3bd3d4c10ad03f1b15a99/action/reboot
```

The content format of Message information is as follows

```
{
    "method": "reboot",
    "msgId": "1651211843741",
    "apiKey": "61a3bd3d4c10ad03f1b15a99",
    "version": 1,
    "message": {
        "mac": ["D4:3D:39:17:42:F6"]
    }
}
```

Message field description

No.	Field	Value Type	Value	Options	Describe

1	method	String	reboot	required	method
2	msgId	String	Timestamp	required	mark unique
4	version	Integer	1	required	version number
5	message.mac	ARRAY	Mac	required	Device MAC address
6	apiKey	String		required	ApiKey

3.13. Client subscribes to device restart feedback result message

The TOPIC format of the client's subscription device restart result message is:

```
/client/${ApiKey}/action/reboot_reply
```

For example, the user ApiKey is 61a3bd3d4c10ad03f1b15a99, and the subscription TOPIC format is as follows

```
/client/61a3bd3d4c10ad03f1b15a99/action/reboot_reply
```

The content format of Message information is as follows

```
{
    "mac": "D4:3D:39:17:42:F6",
    "msgId": "1651211843741",
    "result": 200
}
```

Message field description

No.	Field	Value Type	Value	Describe
1	mac	String		Device MAC address
2	msgId	String	Timestamp	Mark unique
3	result	Integer	200/400	success failure

4. Client HTTP protocol

4.1. HTTP Protocol interface overview

Send setup messages via HTTP protocol

No.	Api	method	Describe
1	/user/api/login	POST	log in system
2	/user/api/rest/devices	GET	Get all device information
3	/user/api/rest/devices/:id	GET	Get individual device information based on id
4	/user/api/rest/devices	POST	Add device

5	/user/api/rest/devices/:id	DELETE	Remove device
6	/user/api/rest/devices/:id	PUT	Update device information
7	/user/api/rest/devices/mac/:mac	GET	Get device based on MAC address
8	/user/api/mqtt/publish/:mac/led	POST	Set device RGB light status
9	/user/api/mqtt/publish/:mac/battery	POST	Get device battery level
10	/user/api/mqtt/publish/:mac/display	POST	Update screen image
11	/user/api/mqtt/publish/:mac/reboot	POST	Reboot the device
12	/user/api/mqtt/publish/{:mac}/template/{:templateId}	POST	Template calling interface

4.2. Log in to the system via HTTP

API interface

```
/user/api/login
```

JAVA Example

```
OkHttpClient client = new OkHttpClient().newBuilder()
  .build();
MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, "{\r\n    \"username\": \"admin\",\r\n
\"password\": \"admin123\"\r\n}");
Request request = new Request.Builder()
  .url("http://192.144.234.153:5000/user/api/login")
  .method("POST", body)
  .addHeader("Content-Type", "application/json")
  .build();
Response response = client.newCall(request).execute();
```

Return results


```
{
  "code": 20000,
  "data": {
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyZjY1NmM2MDVjMTI2NjBk
ZWQyN2JmMyIsImlhdCI6MTY3MTAwOTU1MH0.-nURPZNLx65-
D6wJ1Lgf27Ckr4pLMBLE3_7Cb6BtpaI"
  }
}
```


4.3. Obtain all device information through HTTP

API interface

/user/api/rest/devices

Get the MAC address of all devices

/user/api/rest/devices?query=mac,ip

Return routine


```
{
  "code": 20000,
  "data": {
    "items": [
      {
        "_id": "630333b92f655aed61849d00",
        "mac": "D4:3D:39:1C:E4:CA",
        "ip": "192.168.1.123"
      },
      {
        "_id": "6341242ec977ff50b3d47106",
        "mac": "D4:3D:39:1C:8C:C4",
        "ip": "192.168.1.6"
      },
      {
        "_id": "636091515df509d3871e2f35",
        "mac": "D4:3D:39:1C:64:2C",
        "ip": "192.168.1.110"
      }
    ],
    "total": 3
  }
}
```


Message field description

No.	Field	Value Type	Value	Describe
1	code	Integer		result
2	data.items	Array		Query data array
3	data.total	Integer		Query total value

JAVA Example

```
OkHttpClient client = new OkHttpClient().newBuilder()
  .build();
Request request = new Request.Builder()
  .url("http://192.144.234.153:5000/user/api/rest/devices?query=mac,ip")
  .method("GET", null)
  .addHeader("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyZj
Y1NmM2MDVjMTI2NjBkZWQyN2JmMyIsImlhdCI6MTY3MTAwOTU1MH0.-nURPZNLx65-
D6wJ1Lgf27Ckr4pLMBLE3_7Cb6Btpal")
  .build();
```

```
Response response = client.newCall(request).execute();
```

4.4. Set device RGB light status via HTTP

The URL format is

```
/user/api/mqtt/publish/:mac/led
```

JSON data

```
{
    "red": 0,
    "green": 0,
    "blue": 0,
}
```

JSON Field description

No.	Field	Value Type	Value	Options	Describe
1	red	Integer	0/255	Optional	Red light off/on
2	green	Integer	0/255	Optional	Green light off/on
3	blue	Integer	0/255	Optional	Blue light off/on

Note： : mac is the device MAC address

JAVA Example

```
OkHttpClient client = new OkHttpClient().newBuilder()
  .build();
MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, "{\r\n    \"red\": 255,\r\n    \"green\": 0,\r\n    \"blue\": 0\r\n}");
Request request = new Request.Builder()
  .url("http://192.144.234.153:5000/user/api/mqtt/publish/D4:3D:39:1C:64:2C/led")
  .method("POST", body)
  .addHeader("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyZjY1NmM2MDVjMTI2NjBkZWQyN2JmMyIsImlhdCI6MTY3MTAwOTU1MH0.-nURPZNLx65-D6wJ1Lgf27Ckr4pLMBLE3_7Cb6BtpaI")
  .addHeader("Content-Type", "application/json")
  .build();
Response response = client.newCall(request).execute();
```

Feedback results

```
{
  "code": 20000,
  "data": {
    "msgId": "1671010167925",
```

```
        "mac": "D4:3D:39:1C:64:2C"
    }
}
```

Message field description

No.	Field	Value Type	Value	Describe
1	code	Integer		result
2	data.msgId	String		Message unique identifier
3	data.mac	String		Device MAC address

4.5. Get device battery voltage via HTTP

The URL format is

```
/user/api/mqtt/publish/:mac/battery
```

JSON Data None

Note: :mac is the device MAC address

4.6. Update device screen image via HTTP

The URL format is

```
/user/api/mqtt/publish/:mac/display
```

JSON data

```
{
    "algorithm": "floyd-steinberg",
    "imgsrc": "https://t7.baidu.com/it/u=2141219545,3103086273&fm=193&f=GIF"
}
```

JSON field description

No.	Field	Value Type	Value	Options	Describe
1	algorithm	String		Optional	If not, the system default algorithm
2	imgsrc	String		Optional	1. If not, the system default picture 2. Convert the image to base64 format 3. Image download address

Note: :mac is the device MAC address

JAVA Example

```
OkHttpClient client = new OkHttpClient().newBuilder()
  .build();
MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, "{\r\n\"algorithm\": \"floyd-
steinberg\",\r\n\"imgsrc\": \"https://t7.baidu.com/it/u=2141219545,3103086273&fm=193&f=GI
F\"\r\n}");
Request request = new Request.Builder()
  .url("http://192.144.234.153:5000/user/api/mqtt/publish/D4:3D:39:1C:64:2C/display")
  .method("POST", body)
  .addHeader("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyZj
Y1NmM2MDVjMTI2NjBkZWQyN2JmMyIsImlhdCI6MTY3MTAwOTU1MH0.-nURPZNLx65-
D6wJ1Lgf27Ckr4pLMBLE3_7Cb6BtpaI")
  .addHeader("Content-Type", "application/json")
  .build();
Response response = client.newCall(request).execute();
```

Return results

```
{
  "code": 20000,
  "data": {
    "msgId": "1671010491574",
    "mac": "D4:3D:39:1C:64:2C"
  }
}
```

Message field description

No.	Field	Value Type	Value	Describe
1	code	Integer		result
2	data.msgId	String		Message unique identifier
3	data.mac	String		Device MAC address

4.7. Reboot the device via HTTP

The URL format is

```
/user/api/mqtt/publish/:mac/reboot
```

JSON Data None

Note: :mac is the device MAC address

JAVA Example

```
OkHttpClient client = new OkHttpClient().newBuilder()
  .build();
MediaType mediaType = MediaType.parse("text/plain");
RequestBody body = RequestBody.create(mediaType, "");
```

```
Request request = new Request.Builder()
  .url("http://192.144.234.153:5000/user/api/mqtt/publish/D4:3D:39:1C:64:2C/reboot")
  .method("POST", body)
  .addHeader("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyZj
Y1NmM2MDVjMTI2NjBkZWQyN2JmMyIsImlhdCI6MTY3MTAwOTU1MH0.-nURPZNLx65-
D6wJ1Lgf27Ckr4pLMBLE3_7Cb6BtpaI")
  .build();
Response response = client.newCall(request).execute();
```

Return results

```
{
  "code": 20000,
  "data": {
    "msgId": "1671010699826",
    "mac": "D4:3D:39:1C:64:2C"
  }
}
```

Message field description

No.	Field	Value Type	Value	Describe
1	code	Integer		result
2	data.msgId	String		Message unique identifier
3	data.mac	String		Device MAC address

4.8. Calling the template display interface through HTTP

The URL format is

```
/user/api/mqtt/publish/{:mac}/template/{:templateId}
```

a). mac is the MAC address of the tag, for example: D4:3D:39:1C:8C:C4

b). templateId is the ID number of the template created, for example: 6347e877ac6dda127fb5fd11

JSON data

```
{
        "tid": "6347e877ac6dda127fb5fd11",
        "tname": " Meeting template ",
        "data": {
            " title ": "A2107",
            " URL ": "www.xxxx.cn",
            " content ": " Cheng Xiaoming：  9:00-12:00"
        }
  }
```

JSON Field description

No.	Field	Value Type	Value	Describe
1	tid	String		Template ID
2	tname	String		Template name
3	data	Object		template data

JAVA Example

```
OkHttpClient client = new OkHttpClient().newBuilder()
  .build();
MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, "{\"tid\":\"637212b0c75c505d70807400\",\"tname\":\" test template \",\"data\":{\"text-0\":\" Zhang Sanfeng \",\"text-0-1\":\" Dalian XX Co., Ltd. \"}}");
Request request = new Request.Builder()
  .url("http://192.144.234.153:5000/user/api/mqtt/publish/D4:3D:39:1C:8C:C4/template/637212b0c75c505d70807400")
  .method("POST", body)
  .addHeader("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyZjY1NmM2MDVjMTI2NjBkZWQyN2JmMyIsImlhdCI6MTY2NjI1NjAyOX0.cB4_tWv7i6JK5HGPjCyQ0Yq9NIvGrhS0hll3iqAm9hE")
  .addHeader("Content-Type", "application/json")
  .build();
Response response = client.newCall(request).execute();
```

Return results

```
{
        "code": 20000,
        "data": {
            "mac": "D4:3D:39:1C:8C:C4",
            "msgId": "1666259624740"
        }
    }
```

Message field description

No.	Field	Value Type	Value	Describe
1	code	Integer		result
2	data.msgId	String		Message unique identifier
3	data.mac	String		Device MAC address