2024

WIFI Electronic Tag API Integration Manual

V1.0.6

DALIAN SERTAG TECHNOLOGY CO., LTD

Contents

1. Parameter Config

1.1. WIFI Parameter Config

NO.	Name	Description	Default Value
1	SSID	AP SSID	
2	password	AP password	

1.2. MQTT Parameter Config

NO.	Name	Description	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. Equipment data management

No.	Name	Description	Type	Default Value
1	MAC	MAC address	String	
2	IP	IP address	String	
3	Voltage	Battery power	Integer	
4	Station.RSSI	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	Status	Device status	Boolean	
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	UsbState	USB Status	Integer	
18	Product	Product	Object	18
19	Algorithm	Image Algorithms	Object	19

3. Client and Server MQTT Interaction Protocol

3.1 Overview of the Client MQTT Protocol

NO.	Topic	Description	References
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/${ApiKey}/action/display	Publish device screen update messages	
6	/client/${ApiKey}/action/display_reply	Subscribe to device update screen result messages	
7	/client/${ApiKey}/action/battery	Publish a message to obtain the device battery voltage	
8	/client/${ApiKey}/action/battery_reply	Subscribe to the device battery voltage result message	
9	/client/${ApiKey}/action/led	Post a message updating the LED status	
10	/client/${ApiKey}/action/led_reply	Subscribe to update LED status results	
11	/client/${ApiKey}/action/reboot	Publish restart message	
12	/client/${ApiKey}/action/reboot_reply	Subscribe to the restart result message	

Note: MQTT message content is encoded using base64

The message received is similar to the following

```
ewoJIm1hYyI6ICJENDozRDozOToxNzoyQTo4NCIsCgkibXNnSWQiOiAiMTY1MTIxMTE4NTQ2MiIKfQ==
```

3.2 The Client Subscribes to the Device Online Message

The TOPIC format of the client subscription device online message is

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

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

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

The content format of the Message is as follows

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

Message Field Description

NO.	Field	Type	Value	Description
1	mac	String		MAC address of the device
2	msgId	String	Timestamp	Mark unique

3.3 The client subscribes to the device offline message

The TOPIC format of the client subscription device offline message is as follows:

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

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

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

The content format of the Message is as follows

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

Message Field Description

NO.	Field	Type	Value	Description
1	mac	String		MAC address of the device
2	msgId	String	Timestamp	Mark unique

3.4 The client subscribes to the device USB status feedback message

The client subscribes to the device USB status message TOPIC format is

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

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

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

The content format of the Message is as follows

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

Message Field Description

NO.	Field	Type	Value	Description
1	mac	String		MAC address of the device
2	msgId	String	Timestamp	Mark unique
3	state	Integer	1/0	Insert/remove

3.5 The client subscribes to the device key feedback message

The TOPIC format of the client subscribing to the device key message is as follows:

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

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

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

The content format of the Message is as follows

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

Message Field Description

NO.	Field	Type	Value	Description
1	mac	String		MAC address of the device
2	msgId	String	Timestamp	Mark unique

3.6 The client publishes a message to update the device screen

The TOPIC format of the client's message to update the device screen is

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

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

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

The content format of the Message 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	Type	Value	Options	Description
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	MAC address of the device

6	message.imgsrc	String		Optional	1. If not, the system defaults to the image 2. Convert the image to base64 format 3. Image download address
7	message.template	Object		Optional	Imgsrc or template
8	apiKey	String		Required	ApiKey

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

Updating the screen via templates

```
{
    "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 Client subscription updates the device screen feedback result message

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

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

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

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

The content format of the Message is as follows

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

Message Field Description

NO.	Field	Type	Value	Description
1	mac	String		MAC address of the device
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 client publishes a message to obtain the device battery voltage in the TOPIC format:

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

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

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

The content format of the Message is as follows

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

Message Field Description

NO.	Field	Type	Value	Options	Description
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	MAC address of the device
6	apiKey	String		Required	ApiKey

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

The client subscribes to get the device battery voltage result message TOPIC format is

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

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

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

The content format of the Message is as follows

```
{
    "mac":"D4:3D:39:17:42:F6",
    "msgId":"1651211849119",
    "voltage":410
}
```

Message Field Description

NO.	Field	Type	Value	Description
1	mac	String		MAC address of the device
2	msgId	String	Timestamp	Mark unique
3	voltage	Integer		100 times the battery voltage

3.10 Client publishes update device LED status message

The client publishes an update LED status message TOPIC format is

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

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

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

The content format of the Message 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	Type	Value	Options	Description
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	MAC address of the device
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 Client subscribes to update device LED status feedback result message

The client subscribes to update the LED status result message TOPIC format is

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

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

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

The content format of the Message is as follows

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

Message Field Description

NO.	Field	Type	Value	Description
1	mac	String		MAC address of the device
2	msgId	String	Timestamp	Mark unique
3	result	Integer	200/400	Success/Failure

3.12 The client publishes a device restart message

The client publishes a device restart message in the TOPIC format:

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

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

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

The content format of the Message is as follows

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

Message Field Description

NO.	Field	Type	Value	Options	Description
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	MAC address of the device
6	apiKey	String		Required	ApiKey

3.13 The client subscribes to the device restart feedback result message

The client subscribes to the device restart result message TOPIC format is

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

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

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

The content format of the Message is as follows

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

Message Field Description

NO.	Field	Type	Value	Description
1	mac	String		MAC address of the device
2	msgId	String	Timestamp	Mark unique
3	result	Integer	200/400	Success/Failure

4. Client HTTP protocol

4.1 HTTP Protocol Interface Overview

Sending setup messages via HTTP protocol

No.	API	Method	Description
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 single device information based on id
4	/user/api/rest/devices	POST	Add a device
5	/user/api/rest/devices/:id	DELETE	Deleting a device
6	/user/api/rest/devices/:id	PUT	Update device information
7	/user/api/rest/devices/mac/:mac	GET	Get device by MAC address
8	/user/api/mqtt/publish/:mac/led	POST	Set the 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 call interface

4.2 Login 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://192.144.234.153:4000/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_7Cb6Bt
paI"
    }
}
```

4.3 Get all device information via HTTP

API Interface

```
/user/api/rest/devices
```

Get the MAC addresses of all devices

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

Return to the 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	Type	Value	Description
1	code	Integer		Result
2	data.items	Array		The data array to query
3	data.total	Integer		Query total value

JAVA Example

```
OkHttpClient client = new OkHttpClient().newBuilder()
  .build();
Request request = new Request.Builder()
  .url("http://192.144.234.153:4000/user/api/rest/devices?query=mac,ip")
  .method("GET", null)
  .addHeader("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyZjY1NmM2MDVjMTI2NjBkZWQyN2JmMyIsImlhdCI6MTY3MTAwOTU1MH0.-nURPZNLx65-D6wJ1Lgf27Ckr4pLMBLE3_7Cb6BtpaI")
  .build();
Response response = client.newCall(request).execute();
```

4.4 Set the 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	Type	Value	Options	Description
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://192.144.234.153:4000/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	Type	Value	Description
1	code	Integer		Result
2	data.msgId	String		Message unique identifier
3	data.mac	String		MAC address of the device

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	Type	Value	Options	Description
1	algorithm	String		Optional	If not, the system default algorithm
2	imgsrc	String		Optional	1. If not, the system defaults to the image 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=GIF\"\r\n}");
Request request = new Request.Builder()
.url("http://192.144.234.153:4000/user/api/mqtt/publish/D4:3D:39:1C:64:2C/display")
```

```
        .method("POST", body)
        .addHeader("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyZj
Y1NmM2MDVjMTI2NjBkZWQyN2JmMyIsImlhdCI6MTY3MTAwOTU1MH0.-nURPZNLx65-D6wJ1Lgf
27Ckr4pLMBLE3_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	Type	Value	Description
1	code	Integer		Result
2	data.msgId	String		Message unique identifier
3	data.mac	String		MAC address of the device

4.7 Rebooting a 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://192.144.234.153:4000/user/api/mqtt/publish/D4:3D:39:1C:64:2C/reboot")
      .method("POST", body)
      .addHeader("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyZj
Y1NmM2MDVjMTI2NjBkZWQyN2JmMyIsImlhdCI6MTY3MTAwOTU1MH0.-nURPZNLx65-D6wJ1Lgf
27Ckr4pLMBLE3_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	Type	Value	Description
1	code	Integer		Result
2	data.msgId	String		Message unique identifier
3	data.mac	String		MAC address of the device

4.8 Calling the template display interface via 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 created template, for example, 6347e877ac6dda127fb5fd11JSON data

```
{
    "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"
    }
}
```

JSON Field Description

NO.	Field	Type	Value	Description
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\":\"测试模板\",\"data\":{\"text-0\":\"张三丰\",\"text-0-1\":\"深圳某某有限责任公司\"}}");
Request request = new Request.Builder()
  .url("http://192.144.234.153:4000/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	Type	Value	Description
1	code	Integer		Result
2	data.msgId	String		Message unique identifier
3	data.mac	String		MAC address of the device