AiFace Device MQTT API

Postman request reference · 90 requests

This readable reference accompanies the original collection. Import the JSON file into Postman to use the examples with your own device settings.

Collection settings and usage


```
{
  "info": {
    "name": "AiFace Device MQTT API",
    "_postman_id": "aiface-mqtt-collection-v1",
    "description": "AiFace facial recognition device MQTT/MQTTS communication protocol.\n\n## How to Use\n1. Edit the Collection Variables: set `broker_url` to your MQTT broker address (e.g. `mqtt://192.168.1.100:1883`).\n2. Set `device_sn` to your device serial number.\n3. Open any request folder, click the request, then click **Connect**.\n4. Once connected, click **Send** to publish the command to the device.\n\n## Topic Convention\n- Server publishes commands TO device on: `aiface/<sn>/pub`\n- Device publishes data TO server on: `aiface/<sn>/sub`\n\n## Variables\n- `broker_url` — MQTT broker URL, e.g. `mqtt://192.168.1.100:1883`\n- `device_sn` — Device serial number, e.g. `LF00000001`\n- `enrollid` — Sample user ID used in user management requests",
    "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json"
  },
  "variable": [
    {
      "key": "broker_url",
      "value": "mqtt://192.168.1.100:1883",
      "description": "MQTT Broker URL. Change this to your broker address. Examples:\n  mqtt://192.168.1.100:1883\n mqtts://192.168.1.100:8883"
    },
    {
      "key": "device_sn",
      "value": "LF00000001",
      "description": "Device serial number — must match the SN of the target AiFace device"
    },
    {
      "key": "enrollid",
      "value": "123",
      "description": "Sample user / enrollment ID used in user management requests"
    },
    {
      "key": "pub_topic",
      "value": "aiface/{{device_sn}}/pub",
      "description": "Topic on which the SERVER publishes commands TO the device"
    },
    {
      "key": "sub_topic",
      "value": "aiface/{{device_sn}}/sub",
      "description": "Topic on which the DEVICE publishes data TO the server"
    }
  ]
}
```

4. Device Registration

Commands related to device registration and re-registration. The device sends reg automatically on MQTT connect.

4.1 reg — Device Registration (Device→Server)

MQTT `{{broker_url}}`

The device sends this command automatically immediately after the MQTT connection is established.

Publish this to simulate a device registration message.

Topic: aiface/{{device_sn}}/sub
Direction: Device → Server

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{sub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "reg",
  "sn": "{{device_sn}}",
  "devinfo": {
    "modelname": "AiFace",
    "manufacturer": "AiFace Inc.",
    "usersize": 10000,
    "facesize": 10000,
    "avatarsize": 0,
    "fpsize": 10000,
    "palmsize": 10000,
    "cardsize": 10000,
    "pwdsize": 10000,
    "logsize": 300000,
    "useduser": 100,
    "usedface": 50,
    "usedavatar": 0,
    "usedfp": 20,
    "usedpalm": 0,
    "usedcard": 10,
    "usedpwd": 5,
    "usedlog": 1000,
    "usednewlog": 500,
    "usedrtlog": 3,
    "netinuse": 1,
    "usb4g": 0,
    "fpalgo": "thbio3.0",
    "firmware": "1.0.0",
    "time": "2025-01-01 08:00:00",
```

```
    "intercom": 0,
    "floors": 0,
    "acces_stimes": 0,
    "charid": 0,
    "useosdp": 0,
    "facetemplate": 0,
    "dislanguage": 0,
    "curip": "192.168.001.100",
    "ntp": 1,
    "timezone": "UTC+8",
    "cpu_temp": 45,
    "bandwidth_up": "100Kbps",
    "bandwidth_down": "1Mbps",
    "battery": 100,
    "disksize": 1024,
    "useddisk": 512,
    "mac": "AA-BB-CC-DD-EE-FF"
  }
}
```

4.2 reg — Server Response (Server→Device)

MQTT `{{broker_url}}`

```
Server response to device registration. Publish this to the device command topic.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "ret": "reg",
  "tryseconds": 10,
  "result": true,
  "cloudtime": "2025-01-01 08:00:00",
  "nosenduser": false,
  "nosendlog": false,
  "nosendimage": false
}
```

4.3 getreg — Get Device Registration Info (Server→Device)

MQTT `{{broker_url}}`

```
Retrieves full device registration and capability information.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getreg",
  "sn": "{{device_sn}}"
}
```

5. User Management

```
Commands for managing users: add, update, delete, query user profiles and biometric data.
```

5.1 setuserinfo — Set User Info (Server→Device)

MQTT `{{broker_url}}`

```
Downloads and sets user profile information including biometrics.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "setuserinfo",
  "sn": "{{device_sn}}",
  "enrollid": {{enrollid}},
  "aliasid": "USR001",
  "name": "John Doe",
  "birthday": "01-15",
  "verifymode": 255,
  "card": 12345,
  "pwd": 0,
  "shiftid": 0,
  "zoneid": 0,
  "groupid": 0,
  "access_times": 0,
  "postid": 1,
  "department": "Engineering",
  "enable": 1,
  "starttime": "2025-01-01 00:00:00",
  "endtime": "2025-12-31 23:59:59",
  "backupnum": 50,
  "admin": 0,
  "userprofile": "{}",
  "record": "base64_encoded_face_data"
}
```

5.2 deleteuser — Delete User (Server→Device)

MQTT `{{broker_url}}`

```
Deletes a user or specific biometric data.

backupnum: 12 or 13 = delete entire user, 50 = face photo, 51 = face feature, 10 = password, 11 = card

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "deleteuser",
  "sn": "{{device_sn}}",
  "enrollid": {{enrollid}},
  "aliasid": "USR001",
```

```
  "backupnum": 12
}
```

5.3 cleanuser — Clear All Users (Server→Device)

MQTT `{{broker_url}}`

```
Deletes all users from the device.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "cleanuser",
  "sn": "{{device_sn}}"
}
```

5.4 setusername — Batch Set User Names (Server→Device)

MQTT `{{broker_url}}`

```
Sets names and verify modes for multiple users at once.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "setusername",
  "sn": "{{device_sn}}",
  "count": 2,
  "birthday": "06-15",
  "record": [
    {
      "enrollid": 100,
      "aliasid": "U100",
      "name": "Alice",
      "verifymode": 255
    },
    {
      "enrollid": 101,
      "aliasid": "U101",
      "name": "Bob",
      "verifymode": 255
    }
  ]
}
```

5.5 getusername — Get User Name (Server→Device)

MQTT `{{broker_url}}`

```
Retrieves the name for a specific user.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getusername",
  "sn": "{{device_sn}}",
  "enrollid": {{enrollid}},
  "aliasid": "USR001"
}
```

5.6 getuserlist — Get User List Paginated (Server→Device)

MQTT `{{broker_url}}`

```
Retrieves a paginated list of user IDs. Use stn=1 to reset pagination.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getuserlist",
  "sn": "{{device_sn}}",
  "stn": 1
}
```

5.7 getuserids — Get All User IDs (Server→Device)

MQTT `{{broker_url}}`

```
Returns all user IDs in a single response (no pagination).

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getuserids",
  "sn": "{{device_sn}}"
}
```

5.8 getunuserdid — Get Next Available User ID (Server→Device)

MQTT `{{broker_url}}`

```
Returns the next available unused user ID.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getunuserdid",
  "sn": "{{device_sn}}"
}
```

5.9 checkuserid — Check If User ID Exists (Server→Device)

MQTT `{{broker_url}}`

```
Checks if a specific user ID exists on the device.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "checkuserid",
  "sn": "{{device_sn}}",
```

```
  "enrollid": {{enrollid}}
}
```

5.10 getuserinfo — Get User Info (Server→Device)

MQTT `{{broker_url}}`

```
Retrieves detailed user profile. Omit backupnum for full profile; include backupnum for raw biometric data.

backupnum: 50=face photo, 51=face feature, 10=password, 11=card, 0-9=fingerprint

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getuserinfo",
  "sn": "{{device_sn}}",
  "enrollid": {{enrollid}},
  "aliasid": "USR001"
}
```

5.10b getuserinfo — Get Face Photo (Server→Device)

MQTT `{{broker_url}}`

```
Retrieves face photo (base64 JPG) for a specific user.

backupnum=50 returns face photo data.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
```

```
  }
}
```

Request body

```
{
  "cmd": "getuserinfo",
  "sn": "{{device_sn}}",
  "enrollid": {{enrollid}},
  "aliasid": "USR001",
  "backupnum": 50
}
```

5.11 getallusers — Get All Users Paginated (Server→Device)

MQTT `{{broker_url}}`

```
Returns all user data in paginated format. Each user may return multiple records (one per biometric type). Use
stn=true to start a new query.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getallusers",
  "sn": "{{device_sn}}",
  "stn": true
}
```

5.12 adduser — Start Device-Side Enrollment (Server→Device)

MQTT `{{broker_url}}`

```
Initiates user enrollment on the device. Device will capture biometrics locally.

backupnum: 0-9=fingerprint, 10=password, 11=card, 40/41=palm, 50=face

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "adduser",
  "sn": "{{device_sn}}",
  "enrollid": {{enrollid}},
  "aliasid": "USR001",
  "backupnum": 50,
  "admin": 0,
  "flag": 0,
  "name": "John Doe"
}
```

5.12b adduser — Cancel Enrollment (Server→Device)

MQTT `{{broker_url}}`

```
Cancels the current device-side enrollment process.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "adduser",
  "sn": "{{device_sn}}",
  "cancel": true
}
```

5.13 checkregstatus — Check Enrollment Status (Server→Device)

MQTT `{{broker_url}}`

```
Queries the current enrollment progress on the device.

Response status: 0=waiting, >=0=success (value=backupnum), <0=failure

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "checkregstatus",
  "sn": "{{device_sn}}"
}
```

5.14 canceladduser — Cancel Enrollment (Server→Device)

MQTT `{{broker_url}}`

```
Cancels the current user enrollment process.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "canceladduser",
```

```
  "sn": "{{device_sn}}"
}
```

5.15 enableuser — Enable/Disable User (Server→Device)

MQTT `{{broker_url}}`

```
Enables or disables a user. Set enflag=true to enable, false to disable.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "enableuser",
  "sn": "{{device_sn}}",
  "enrollid": {{enrollid}},
  "aliasid": "USR001",
  "enflag": true
}
```

5.16 getuserprofile — Get User Custom Profile (Server→Device)

MQTT `{{broker_url}}`

```
Retrieves the custom profile JSON string for a user.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getuserprofile",
  "sn": "{{device_sn}}",
  "enrollid": {{enrollid}},
  "aliasid": "USR001"
}
```

5.17 setuserprofile — Set User Custom Profile (Server→Device)

MQTT `{{broker_url}}`

```
Sets the custom profile JSON string for a user.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "setuserprofile",
  "sn": "{{device_sn}}",
  "enrollid": {{enrollid}},
  "aliasid": "USR001",
  "profile": "{\"key\": \"value\"}",
  "duration": 5,
  "color": 1
}
```

6. Log Management

```
Commands for retrieving, filtering, and clearing attendance and access logs.
```

6.1 getnewlog — Get New/Unread Logs (Server→Device)

MQTT `{{broker_url}}`

```
Retrieves new (unread) attendance/access log records with pagination.
```

```
Use stn=1 to reset pagination. Set logphoto=1 to include user photos.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getnewlog",
  "sn": "{{device_sn}}",
  "stn": 1,
  "logphoto": 0
}
```

6.2 getalllog — Get All Logs (Server→Device)

MQTT `{{broker_url}}`

```
Retrieves all log records with optional filtering by user and date range.

Date format: YYYY-MM-DD (e.g. 2025-01-01)

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getalllog",
  "sn": "{{device_sn}}",
  "stn": 1,
  "from": "2025-01-01",
  "to": "2025-12-31",
```

```
  "logphoto": 0
}
```

6.3 cleanlog — Clear All Logs (Server→Device)

MQTT `{{broker_url}}`

```
Deletes all log records from the device.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "cleanlog",
  "sn": "{{device_sn}}"
}
```

6.4 cleanlogphoto — Clear Log Photos (Server→Device)

MQTT `{{broker_url}}`

```
Deletes all photos associated with log records.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "cleanlogphoto",
  "sn": "{{device_sn}}"
}
```

7. Device Management

```
Commands for reading and setting device parameters, time, system reset, and device state control.
```

7.1 getdevinfo — Get Device Parameters (Server→Device)

MQTT `{{broker_url}}`

```
Reads all system parameters. Returns all DI_FIRST to DI_END parameters as key-value pairs.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getdevinfo",
  "sn": "{{device_sn}}"
}
```

7.2 setdevinfo — Set Device Parameters (Server→Device)

MQTT `{{broker_url}}`

```
Sets one or more system parameters. Adding DI_SERVERIP/DI_SERVERPORT will trigger connection re-establishment.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "setdevinfo",
  "sn": "{{device_sn}}",
  "nowtime": 1735689600,
  "DI_SERVERIP": "192.168.1.100",
  "DI_SERVERPORT": 81
}
```

7.3 getdevcap — Get Device Capacity (Server→Device)

MQTT `{{broker_url}}`

```
Returns device storage capacity information for all biometric types.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getdevcap",
  "sn": "{{device_sn}}"
}
```

7.4 gettime — Get Device Time (Server→Device)

MQTT `{{broker_url}}`

```
Gets the current device time.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "gettime",
  "sn": "{{device_sn}}"
}
```

7.5 settime — Set Device Time (Server→Device)

MQTT `{{broker_url}}`

```
Sets the device time. Provide both cloudtime (string) and nowtime (Unix ms timestamp).

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "settime",
  "sn": "{{device_sn}}",
  "cloudtime": "2025-01-01 08:00:00",
  "nowtime": 1735689600
}
```

7.6 initsys — Factory Reset (Server→Device)

MQTT `{{broker_url}}`

```
Performs a factory reset: clears all users and logs, restores default settings, backs up WiFi configuration.

WARNING: This will erase all user data!

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "initsys",
  "sn": "{{device_sn}}"
}
```

7.7 initmenu — Reset Menu Settings (Server→Device)

MQTT `{{broker_url}}`

```
Restores default menu settings. Preserves language and backs up WiFi configuration.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "initmenu",
```

```
  "sn": "{{device_sn}}"
}
```

7.8 cleandatebase — Clear Database (Server→Device)

MQTT `{{broker_url}}`

```
Clears the entire database. Device must be rebooted after this operation.

WARNING: Requires reboot after execution!

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "cleandatebase",
  "sn": "{{device_sn}}"
}
```

7.9 cleanadmin — Clear All Administrators (Server→Device)

MQTT `{{broker_url}}`

```
Removes all administrator privileges from the device.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "cleanadmin",
  "sn": "{{device_sn}}"
}
```

7.10 cleaninactiveuser — Clear Inactive Users (Server→Device)

MQTT `{{broker_url}}`

```
Removes users who have not been active (no attendance records).

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "cleaninactiveuser",
  "sn": "{{device_sn}}"
}
```

7.11 reboot — Reboot Device (Server→Device)

MQTT `{{broker_url}}`

```
Triggers an immediate device reboot. No response is sent — device restarts immediately.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
```

```
  }
}
```

Request body

```
{
  "cmd": "reboot",
  "sn": "{{device_sn}}"
}
```

7.12 disabledevice — Disable Device Recognition (Server→Device)

MQTT `{{broker_url}}`

```
Pauses all face, card, fingerprint, and palm detection.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "disabledevice",
  "sn": "{{device_sn}}"
}
```

7.13 enabledevice — Enable Device Recognition (Server→Device)

MQTT `{{broker_url}}`

```
Resumes all face, card, fingerprint, and palm detection.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
```

```
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "enabledevice",
  "sn": "{{device_sn}}"
}
```

7.14 checklive — Heartbeat Check (Server→Device)

MQTT `{{broker_url}}`

```
Server-initiated heartbeat to verify device connectivity. Device responds to confirm it is alive.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "checklive",
  "sn": "{{device_sn}}"
}
```

8. Access Control

```
Commands for door control, lock configuration, and user access parameters.
```

8.1 opendoor — Remote Door Open (Server→Device)

MQTT `{{broker_url}}`

```
Remotely unlocks the door. Optionally sends user ID via Wiegand and displays a message.
```

```
Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "opendoor",
  "sn": "{{device_sn}}",
  "enrollid": {{enrollid}},
  "floor": 0,
  "msg": "Welcome"
}
```

8.2 lockctrl — Door Lock Control (Server→Device)

MQTT `{{broker_url}}`

```
Controls the door lock state directly.

fuc: 1=unlock/open, 2=lock/close

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "lockctrl",
  "sn": "{{device_sn}}",
  "fuc": 1
}
```

8.3 getdoorstatus — Get Door Status (Server→Device)

MQTT `{{broker_url}}`

```
Returns the current door sensor status.

doorstatus in response: 0=closed, 1=open

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getdoorstatus",
  "sn": "{{device_sn}}"
}
```

8.4 setdevlock — Set Door Access Parameters (Server→Device)

MQTT `{{broker_url}}`

```
Configures door access control parameters including timing, zones, and Wiegand settings.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "setdevlock",
  "sn": "{{device_sn}}",
```

```
  "opendelay": 5,
  "doorsensor": 0,
  "alarmdelay": 30,
  "InputAlarm": 0,
  "antpass": 0,
  "interlock": 0,
  "mutiopen": 0,
  "tryalarm": 5,
  "access_times": 0,
  "punchtimes": 0,
  "tamper": 0,
  "wgformat": 0,
  "wgoutput": 0,
  "cardformat": 0,
  "wgport": 0
}
```

8.5 getdevlock — Get Door Access Parameters (Server→Device)

MQTT `{{broker_url}}`

```
Reads the current door access control configuration.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getdevlock",
  "sn": "{{device_sn}}"
}
```

8.6 getuserlock — Get User Door Access Parameters (Server→Device)

MQTT `{{broker_url}}`

```
Reads door access parameters for a specific user.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getuserlock",
  "sn": "{{device_sn}}",
  "enrollid": {{enrollid}},
  "aliasid": "USR001"
}
```

8.7 setuserlock — Batch Set User Access Parameters (Server→Device)

MQTT `{{broker_url}}`

```
Sets door access time zones and restrictions for multiple users at once.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "setuserlock",
  "sn": "{{device_sn}}",
  "count": 2,
  "birthday": "06-15",
  "record": [
    {
      "enrollid": 100,
      "aliasid": "U100",
      "verifymode": 255,
      "weekzone": 0,
      "group": 0,
      "access_times": 0,
      "starttime": "2025-01-01 00:00:00",
```

```
      "endtime": "2025-12-31 23:59:59"
    },
    {
      "enrollid": 101,
      "aliasid": "U101",
      "verifymode": 255,
      "weekzone": 1,
      "group": 1,
      "access_times": 10,
      "starttime": "2025-01-01 00:00:00",
      "endtime": "2025-12-31 23:59:59"
    }
  ]
}
```

8.8 deleteuserlock — Delete User Access Parameters (Server→Device)

MQTT `{{broker_url}}`

```
Removes door access parameters for a specific user.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "deleteuserlock",
  "sn": "{{device_sn}}",
  "enrollid": {{enrollid}},
  "aliasid": "USR001"
}
```

8.9 cleanuserlock — Clear All User Access Parameters (Server→Device)

MQTT `{{broker_url}}`

```
Removes door access time zones and restrictions for all users.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "cleanuserlock",
  "sn": "{{device_sn}}"
}
```

9. Attendance Management

```
Commands for shift schedules, bell times, and holiday configurations.
```

9.1 getshift — Get Shift Schedule (Server→Device)

MQTT `{{broker_url}}`

```
Reads the configured shift/roster schedule. Max 8 shifts, each with up to 3 time sections.

Section type: 0=work, 1=rest, 2=overtime

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getshift",
  "sn": "{{device_sn}}"
}
```

9.2 setshift — Set Shift Schedule (Server→Device)

MQTT `{{broker_url}}`

```
Configures the shift/roster schedule.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "setshift",
  "sn": "{{device_sn}}",
  "shifts": [
    {
      "name": "Day Shift",
      "cutoftime": "09:00",
      "sections": [
        {
          "start": "08:00",
          "end": "12:00",
          "type": 0
        },
        {
          "start": "13:00",
          "end": "17:00",
          "type": 1
        }
      ]
    },
    {
      "name": "Night Shift",
      "cutoftime": "21:00",
      "sections": [
        {
          "start": "20:00",
          "end": "23:59",
          "type": 0
        },
        {
          "start": "00:00",
          "end": "06:00",
          "type": 0
        }
      ]
    }
  ]
}
```

9.3 getbelltime — Get Bell Schedule (Server→Device)

MQTT `{{broker_url}}`

```
Reads configured bell/doorbell time schedules.

flag is a bitmask where each bit = a day of week (127 = all days)

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getbelltime",
  "sn": "{{device_sn}}"
}
```

9.4 setbelltime — Set Bell Schedule (Server→Device)

MQTT `{{broker_url}}`

```
Configures bell/doorbell time schedules.

flag: bitmask for days of week (127=all days, 65=Mon+Sun, etc.)

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "setbelltime",
  "sn": "{{device_sn}}",
  "belltimes": [
    {
      "time": "08:00",
      "flag": 127
    },
    {
      "time": "12:00",
      "flag": 127
    },
    {
      "time": "18:00",
      "flag": 65
    }
  ]
}
```

9.5 setholiday — Set Holiday Schedule (Server→Device)

MQTT `{{broker_url}}`

```
Configures holiday schedules.

Date format: MM-DD

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "setholiday",
  "sn": "{{device_sn}}",
  "holidays": [
    {
      "name": "Spring Festival",
      "startday": "01-28",
      "endday": "02-03",
      "shift": 0,
      "dayzone": 0
    },
    {
      "name": "National Day",
```

```
      "startday": "10-01",
      "endday": "10-07",
      "shift": 1,
      "dayzone": 0
    }
  ]
}
```

9.6 getholiday — Get Holiday Schedule (Server→Device)

MQTT `{{broker_url}}`

```
Reads configured holiday schedule.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getholiday",
  "sn": "{{device_sn}}"
}
```

10. Device Active Reports (Device→Server)

```
Commands initiated by the device and sent to the server. The server processes them and sends back a response.

These are device-to-server messages. Use them to simulate device behavior for testing.
```

10.1 sendlog — Device Reports Attendance Logs (Device→Server)

MQTT `{{broker_url}}`

```
Device proactively sends attendance/access log records to the server.

Publish to sub_topic to simulate device sending logs.

Topic: aiface/{{device_sn}}/sub
Direction: Device → Server
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{sub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "sendlog",
  "sn": "{{device_sn}}",
  "count": 1,
  "logindex": 0,
  "record": [
    {
      "enrollid": {{enrollid}},
      "name": "John Doe",
      "time": "2025-01-01 08:00:00",
      "mode": 1,
      "inout": 0,
      "event": 0,
      "aliasid": "USR001",
      "image": "",
      "note": ""
    }
  ]
}
```

10.2 senduser — Device Reports User Data Change (Device→Server)

MQTT {{broker_url}}

```
Device proactively sends user enrollment/modification data when biometrics are enrolled on the device.

Example: face photo upload (backupnum=50)

Topic: aiface/{{device_sn}}/sub
Direction: Device → Server
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{sub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "senduser",
  "sn": "{{device_sn}}",
  "enrollid": {{enrollid}},
  "name": "John Doe",
  "backupnum": 50,
  "admin": 0,
  "record": "base64_jpg_data",
  "aliasid": "USR001"
}
```

10.3 sendpin — Device Reports PIN Input (Device→Server)

MQTT `{{broker_url}}`

```
Device reports when a user enters a PIN code on the device keypad.

Topic: aiface/{{device_sn}}/sub
Direction: Device → Server
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{sub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "sendpin",
  "sn": "{{device_sn}}",
  "pin": "123456",
  "time": "2025-01-01 08:00:00"
}
```

10.4 sendqrcode — Device Reports QR Code Scan (Device→Server)

MQTT `{{broker_url}}`

```
Device reports when a user scans a QR code.

Topic: aiface/{{device_sn}}/sub
Direction: Device → Server
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{sub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "sendqrcode",
  "sn": "{{device_sn}}",
  "record": "qrcode_data_string"
}
```

10.5 sendgps — Device Reports GPS Location (Device→Server)

MQTT `{{broker_url}}`

```
Device proactively sends GPS location data (requires USB 4G module).

Topic: aiface/{{device_sn}}/sub
Direction: Device → Server
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{sub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "sendgps",
  "sn": "{{device_sn}}",
  "satellites": 8,
  "location": "22.5431,114.0579",
  "timeStamp": 1704067200
}
```

10.6 otacheck — OTA Update Check (Device→Server)

MQTT `{{broker_url}}`

```
Device checks if a firmware update is available.

Topic: aiface/{{device_sn}}/sub
Direction: Device → Server
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{sub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "otacheck",
  "sn": "{{device_sn}}",
  "fwname": "ai230_device",
  "version": "1.0.0"
}
```

10.7 otaget — OTA File Download Chunk (Device→Server)

MQTT `{{broker_url}}`

```
Device requests a chunk of the OTA firmware file.

Topic: aiface/{{device_sn}}/sub
Direction: Device → Server
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{sub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "otaget",
  "sn": "{{device_sn}}",
  "filename": "update.pkg",
  "index": 0,
  "count": 102400
}
```

11. File Management

```
Commands for listing, reading, and writing files on the device.
```

11.1 getdir — List Directory Files (Server→Device)

MQTT `{{broker_url}}`

```
Lists files in a specified directory. Path must start with /data/

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getdir",
  "sn": "{{device_sn}}",
  "path": "/data/photos"
}
```

11.2 getfile — Read File (Server→Device)

MQTT `{{broker_url}}`

```
Reads the contents of a file and returns it as base64.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getfile",
  "sn": "{{device_sn}}",
  "path": "/data/photos/user123.jpg"
}
```

11.3 writefile — Write File (Server→Device)

MQTT `{{broker_url}}`

```
Writes base64-encoded content to a file on the device.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "writefile",
  "sn": "{{device_sn}}",
  "path": "/data/config/settings.json",
  "base64": "base64_encoded_json_data"
}
```

12. Other Commands

```
Firmware upgrade, OTA server configuration, screensaver, voice, keypad, biometric verification, department,
company, and questionnaire commands.
```

12.1 upgrade — Firmware Upgrade via Server (Server→Device)

MQTT `{{broker_url}}`

```
Uploads firmware upgrade data to the device in chunks.

This is a multi-chunk operation. Send chunk by chunk, incrementing the index each time.
```

```
Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "upgrade",
  "sn": "{{device_sn}}",
  "filename": "update.pkg",
  "md5": "abc123def456",
  "allcount": 10485760,
  "index": 0,
  "count": 102400,
  "record": "base64_encoded_chunk_data"
}
```

12.2 forceota — Force OTA Update (Server→Device)

MQTT `{{broker_url}}`

```
Triggers an immediate OTA update check and download.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "forceota",
  "sn": "{{device_sn}}"
}
```

12.3 setotaserver — Set OTA Server Address (Server→Device)

MQTT `{{broker_url}}`

```
Sets the OTA server address.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "setotaserver",
  "sn": "{{device_sn}}",
  "address": "https://ota.example.com"
}
```

12.4 getotaserver — Get OTA Server Address (Server→Device)

MQTT `{{broker_url}}`

```
Reads the configured OTA server address.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getotaserver",
```

```
  "sn": "{{device_sn}}"
}
```

12.5 setscreensaver — Set Screensaver (Server→Device)

MQTT `{{broker_url}}`

```
Configures the device screensaver with an image.

Set removeall=true to delete all screensavers. Set remove=true to delete a specific file.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "setscreensaver",
  "sn": "{{device_sn}}",
  "filename": "screensaver1.jpg",
  "photo": "base64_encoded_image",
  "duration": 30,
  "remove": false,
  "removeall": false
}
```

12.6 setvoice — Set Custom Voice (Server→Device)

MQTT `{{broker_url}}`

```
Uploads and optionally plays a custom WAV voice file.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
```

```
  }
}
```

Request body

```
{
  "cmd": "setvoice",
  "sn": "{{device_sn}}",
  "index": 10,
  "voice": "base64_encoded_wav_data",
  "play": true,
  "repeat_seconds": 60
}
```

12.7 keypad — Control On-Screen Keypad (Server→Device)

MQTT `{{broker_url}}`

```
Activates or deactivates the on-screen numeric keypad.

on=true to activate, on=false to deactivate

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "keypad",
  "sn": "{{device_sn}}",
  "password": true,
  "keep": false,
  "timeout": 30,
  "on": true
}
```

12.8 verify — Biometric Verification (Server→Device)

MQTT `{{broker_url}}`

```
Performs face or fingerprint verification on the server side.

Provide face (base64 JPG) or finger (hex with compression) for verification.
```

```
Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "verify",
  "sn": "{{device_sn}}",
  "enrollid": {{enrollid}},
  "card": 12345,
  "pwd": 0,
  "name": "John Doe",
  "face": "base64_face_photo",
  "finger": "",
  "finger2": "",
  "profile": "{}"
}
```

12.9 setdepartment — Set Department List (Server→Device)

MQTT `{{broker_url}}`

```
Sets the department list on the device. Set delete=true to clear all departments.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "setdepartment",
  "sn": "{{device_sn}}",
```

```
  "delete": false,
  "name": [
    "Engineering",
    "Sales",
    "HR",
    "Finance",
    "Operations"
  ]
}
```

12.10 getdepartment — Get Department List (Server→Device)

MQTT `{{broker_url}}`

```
Reads the configured department list.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getdepartment",
  "sn": "{{device_sn}}"
}
```

12.11 setcompanyname — Set Company Name (Server→Device)

MQTT `{{broker_url}}`

```
Sets the company name displayed on the device.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
```

```
  }
}
```

Request body

```
{
  "cmd": "setcompanyname",
  "sn": "{{device_sn}}",
  "name": "Acme Corporation"
}
```

12.12 getcompanyname — Get Company Name (Server→Device)

MQTT `{{broker_url}}`

```
Reads the configured company name.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "getcompanyname",
  "sn": "{{device_sn}}"
}
```

12.13 setquestionnaire — Set Questionnaire (Server→Device)

MQTT `{{broker_url}}`

```
Configures a questionnaire/survey displayed to users after verification.

Schedule format: HH:MM-HH:MM*N (N = repeat interval in minutes)

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "setquestionnaire",
  "sn": "{{device_sn}}",
  "title": "Health Declaration",
  "voice": "health.wav",
  "errmsg": "Please complete the survey",
  "radio": true,
  "optionflag": 0,
  "items": [
    "Option A",
    "Option B",
    "Option C"
  ],
  "schedules": [
    "08:00-18:00*1"
  ],
  "usequestion": true,
  "useschedule": false,
  "card": 0,
  "switch_card": 0,
  "enrollid": 0,
  "open": false
}
```

13. Video Intercom Commands

13.1 enablewebrtc — Enable WebRTC (Server→Device)

MQTT `{{broker_url}}`

```
Enable / configure the WebRTC video intercom (intercom & monitor modes). Requires hardware support.

inputmode: 0=direct call (no input dialog), 1=enroll/ID number, 2=phone number, 3=room number, 4=visual selection
menu.
audioonly: number / boolean — 1/true = audio only, 0/false = audio + video (also controls video on/off).

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "enablewebrtc",
  "sn": "{{device_sn}}",
  "intercom": true,
  "monitor": true,
  "waitseconds": 30,
  "inputmode": 0,
  "audioonly": false,
  "stunserver": "stun:stun.example.com:3478",
  "turnserver": "turn:turn.example.com:3478",
  "turnusername": "user",
  "turnpassword": "password"
}
```

13.2 callaccept — Accept or Reject Call (Server→Device)

MQTT `{{broker_url}}`

```
Accept or reject an incoming intercom call. result=true accepts, result=false rejects.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "callaccept",
  "sn": "{{device_sn}}",
  "result": true,
  "sessionid": "session_abc123"
}
```

13.3 viewready — Monitor Ready (Server→Device)

MQTT `{{broker_url}}`

```
Tells the device that the monitoring stream is ready to be viewed.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "viewready",
  "sn": "{{device_sn}}",
  "sessionid": "session_abc123"
}
```

13.4 answer — Process WebRTC Answer (Server→Device)

MQTT `{{broker_url}}`

```
Delivers the WebRTC SDP answer to the device for the given session.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "answer",
  "sn": "{{device_sn}}",
  "sessionid": "session_abc123",
```

```
  "sdp": "v=0\r\no=..."
}
```

13.5 icecandidate — Process ICE Candidate (Server→Device)

MQTT `{{broker_url}}`

```
Delivers a WebRTC ICE candidate to the device for the given session.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "icecandidate",
  "sn": "{{device_sn}}",
  "sessionid": "session_abc123",
  "candidate": "candidate:1 1 UDP 123456 192.168.1.100 12345 typ host"
}
```

13.6 callcancel — Cancel Call (Server→Device)

MQTT `{{broker_url}}`

```
Cancels / hangs up the intercom or monitor session.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "callcancel",
  "sn": "{{device_sn}}",
  "sessionid": "session_abc123",
  "type": 1
}
```

13.7 talklock — Door Control During Call (Server→Device)

MQTT `{{broker_url}}`

```
Unlocks the door while an intercom call is in progress. type: 1 = unlock.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "talklock",
  "sn": "{{device_sn}}",
  "type": 1
}
```

13.8 monitorcall — Start Monitoring (Server→Device)

MQTT `{{broker_url}}`

```
Asks the device to start a monitoring (view) session. force=true forces the session even if another user is
monitoring.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
```

```
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "monitorcall",
  "sn": "{{device_sn}}",
  "force": false,
  "sessionid": "session_abc123"
}
```

13.9 setviselection — Visual Selection Menu, FLAT mode (Server→Device)

MQTT `{{broker_url}}`

```
Configure the interactive visual selection menu shown on the device touchscreen — FLAT mode.

Each page uses items[] ONLY (mutually exclusive with floors[] — do not send both). Max 6 pages, max 500 items per page.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "setviselection",
  "sn": "{{device_sn}}",
  "title": "Visitor Menu",
  "voice": "Please select a room",
  "pages": [
    {
      "itemtitle": "Building 1",
      "items": [
        {
          "section": "101",
          "keynum": 101
        },
        {
          "section": "102",
          "keynum": 102
        }
      ]
```

```
    }
  ]
}
```

13.9b setviselection — Visual Selection Menu, FLOOR mode (Server→Device)

MQTT `{{broker_url}}`

```
Configure the visual selection menu — FLOOR mode (building → floor → room).

Each page uses floors[] ONLY (mutually exclusive with items[] — do not send both). Max 6 pages, max 100 floors per page, max 50 rooms per floor.

Topic: aiface/{{device_sn}}/pub
Direction: Server → Device
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{pub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "setviselection",
  "sn": "{{device_sn}}",
  "title": "Visitor Menu",
  "voice": "Please select a floor",
  "pages": [
    {
      "itemtitle": "Building 1",
      "floors": [
        {
          "floortitle": "Floor 1",
          "items": [
            {
              "section": "101",
              "keynum": 101
            },
            {
              "section": "102",
              "keynum": 102
            }
          ]
        },
        {
          "floortitle": "Floor 2",
          "items": [
            {
              "section": "201",
              "keynum": 201
            }
          ]
```

```
        }
      ]
    }
  ]
}
```

13.10 talkcall — Device Initiates Call (Device→Server)

MQTT `{{broker_url}}`

```
Device proactively initiates an intercom call to the server / subscriber.

Topic: aiface/{{device_sn}}/sub
Direction: Device → Server
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{sub_topic}}",
    "qos": "1",
    "retain": false
  }
}
```

Request body

```
{
  "cmd": "talkcall",
  "sn": "{{device_sn}}",
  "inputnum": 0,
  "inputmode": 0,
  "landscape": 1
}
```

13.11 vicheckonline — Check Online Status (Device→Server)

MQTT `{{broker_url}}`

```
Device checks whether the subscribers / indoor units are online.

Topic: aiface/{{device_sn}}/sub
Direction: Device → Server
```

Request configuration

```
{
  "header": [],
  "mqtt": {
    "topic": "{{sub_topic}}",
    "qos": "1",
    "retain": false
```

```
  }
}
```

Request body

```
{
  "cmd": "vicheckonline",
  "sn": "{{device_sn}}"
}
```