AiFace Device Web API

Postman request reference · 81 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": {
    "_postman_id": "aiface-device-api-collection",
    "name": "AiFace Device Web API",
    "description": "AiFace Facial Recognition Access Control Device - Complete HTTP JSON API Collection\n\n## Usage\n1. Import this collection into Postman\n2. Go to Variables tab (Collection-level Variables)\n3. Set base_url to your device IP (default: http://192.168.1.224)\n4. Set password to your device password (default: 12345678)\n5. All requests will automatically use your configured values\n\n---\n\nProtocol: HTTP + JSON | Base URL: {{base_url}} | Main Endpoint: POST /api | Port: 80 (default) | Auth: All commands require password field except getlang",
    "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json"
  },
  "variable": [
    {
      "key": "base_url",
      "value": "http://192.168.1.224",
      "type": "string"
    },
    {
      "key": "password",
      "value": "12345678",
      "type": "string"
    },
    {
      "key": "enrollid",
      "value": "1001",
      "type": "string"
    }
  ]
}
```


7.1 Authentication & Device Info

getlang - Get Current Language

POST `{{base_url}}/api`

```
No password required. Returns current device language.
```

Request configuration


```
{
  "header": [
```

```
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getlang"
}
```

Examples and scripts

```
{
  "event": [
    {
      "listen": "test",
      "script": {
        "exec": [
          "// Response check",
          "if (pm.response.json().result === true) {",
          "    pm.test(\"Language: \" + pm.response.json().lang);",
          "}"
        ],
        "type": "text/javascript"
      }
    }
  ]
}
```

getlangoption - Get Language List

POST {{base_url}}/api

```
Get available language list enabled in device settings.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getlangoption",
```

```
  "password": "{{password}}"
}
```

login - Login Verification

POST {{base_url}}/api

```
Verify admin login credentials.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "login",
  "username": "admin",
  "password": "{{password}}"
}
```

reg - Get Full Device Registration Info

POST {{base_url}}/api

```
Returns complete device info: model, firmware, capacity, network, storage, biometric counts, etc.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "reg",
  "password": "{{password}}"
}
```

getdevinfo - Get Device Configuration Parameters

POST {{base_url}}/api

```
Returns all DI_* configuration key-value pairs (network, door, biometric, server settings, etc.)
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getdevinfo",
  "password": "{{password}}"
}
```

setdevinfo - Set Device Configuration

POST {{base_url}}/api

```
Set any DI_* configuration parameters. Only include fields you want to change. Supports time sync via nowtime (Unix timestamp ms).
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "setdevinfo",
  "password": "{{password}}",
  "nowtime": 1700000000,
  "deviceid": 1,
  "volume": 80,
  "live_detect": 1,
  "fcmatch_level": 60
}
```

getdevcap - Get Device Capacity Info

POST `{{base_url}}/api`

```
Returns max capacities and current usage for users, faces, fingerprints, palms, cards, passwords, logs.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getdevcap",
  "password": "{{password}}"
}
```

7.2 User Management

getuserlist - Get User List (Paginated)

POST `{{base_url}}/api`

```
Get paginated user list. stn=1 to start from beginning, stn=0 to continue next page. Max 50 records per page.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getuserlist",
  "password": "{{password}}",
  "stn": 1
}
```

Examples and scripts

```
{
  "event": [
    {
      "listen": "test",
      "script": {
        "exec": [
          "let json = pm.response.json();",
          "pm.test('User count: ' + json.count, function() { pm.expect(json.result).to.eql(true); });"
        ],
        "type": "text/javascript"
      }
    }
  ]
}
```

getuserids - Get All User IDs

POST `{{base_url}}/api`

```
Get array of all enrolled user ID strings.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getuserids",
  "password": "{{password}}"
}
```

getunuserdid - Get Next Available User ID

POST `{{base_url}}/api`

```
Returns the next unused user ID for new user enrollment.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
```

```
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getunuserdid",
  "password": "{{password}}"
}
```

checkuserid - Check If User ID Exists

POST `{{base_url}}/api`

```
Check whether a specific user ID exists in the device.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "checkuserid",
  "password": "{{password}}",
  "enrollid": "{{enrollid}}"
}
```

getuserinfo - Get Single User Details (+Face Photo)

POST `{{base_url}}/api`

```
Get detailed user info. backupnum=50 includes Base64 face photo. See Appendix for other backupnum values.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
```

```
  ]
}
```

Request body

```
{
  "cmd": "getuserinfo",
  "password": "{{password}}",
  "enrollid": "{{enrollid}}",
  "backupnum": 50
}
```

setuserinfo - Add/Update User (+Face)

POST `{{base_url}}/api`

```
Add new user or update existing user. Include base64 face JPEG in face field for face enrollment. Max 512KB.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "setuserinfo",
  "password": "{{password}}",
  "enrollid": "{{enrollid}}",
  "name": "John Doe",
  "admin": 0,
  "pwd": 123456,
  "card": 12345678,
  "department": "Engineering",
  "birthday": "01-15",
  "starttime": "2024-01-01 00:00:00",
  "endtime": "2025-12-31 23:59:59",
  "verifymode": 0,
  "shiftid": 1,
  "zoneid": 1,
  "groupid": 0,
  "access_times": 0,
  "enable": 1,
  "face": "<Base64-encoded face JPEG>"
}
```

deleteuser - Delete User/Biometric Data

POST `{{base_url}}/api`

```
Delete user or specific biometric data. Use backupnum to select action type.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "deleteuser",
  "password": "{{password}}",
  "enrollid": "{{enrollid}}",
  "backupnum": 12
}

/* backupnum action table:
 * 12 or 13 = Delete entire user
 * 0-9      = Delete fingerprint at index
 * 10       = Clear password
 * 11       = Clear card
 * 40/41    = Delete left/right palm print
 * 50/51    = Delete face
 */
```

deleteuserface - Delete User Face

POST `{{base_url}}/api`

```
Delete face data for a specific user.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "deleteuserface",
  "password": "{{password}}",
```

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

deleteuserfp - Delete All User Fingerprints

POST `{{base_url}}/api`

```
Delete all fingerprint data for a specific user.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "deleteuserfp",
  "password": "{{password}}",
  "enrollid": "{{enrollid}}"
}
```

deleteuserpalm - Delete All User Palm Prints

POST `{{base_url}}/api`

```
Delete all palm print data for a specific user.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "deleteuserpalm",
  "password": "{{password}}",
  "enrollid": "{{enrollid}}"
}
```

deleteusers - Batch Delete Users

POST {{base_url}}/api

```
Batch delete multiple users by ID array.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "deleteusers",
  "password": "{{password}}",
  "list": [
    "1001",
    "1002",
    "1003"
  ]
}
```

cleanuser - Delete ALL Users [!]

POST {{base_url}}/api

```
[!] DESTRUCTIVE: Removes ALL enrolled users from the device.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "cleanuser",
  "password": "{{password}}"
}
```

enableuser - Enable/Disable User

POST `{{base_url}}/api`

```
Enable (1) or disable (0) a specific user account.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "enableuser",
  "password": "{{password}}",
  "enrollid": "{{enrollid}}",
  "enflag": 1
}

/* enflag: 1 = enable, 0 = disable */
```

7.3 Biometric Enrollment (Device-Side)

adduser - Trigger On-Device Enrollment

POST `{{base_url}}/api`

```
Triggers on-device biometric enrollment screen. Use cancel:true to cancel. Check status with checkregstatus.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "adduser",
```

```
  "password": "{{password}}",
  "enrollid": "{{enrollid}}",
  "name": "John Doe",
  "admin": 0,
  "backupnum": 50,
  "flag": 0
}
```

adduser - Cancel Enrollment

POST {{base_url}}/api

```
Cancel an ongoing on-device enrollment process.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "adduser",
  "password": "{{password}}",
  "cancel": true
}
```

checkregstatus - Query Enrollment Status

POST {{base_url}}/api

```
Query on-device enrollment status. -1=idle, 0=in progress, 1=success, other=failed.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "checkregstatus",
  "password": "{{password}}"
}
```

Examples and scripts

```
{
  "event": [
    {
      "listen": "test",
      "script": {
        "exec": [
          "let json = pm.response.json();",
          "let statusMap = {'-1': 'Idle/cancelled', '0': 'In progress', '1': 'Success'};",
          "pm.test('Status: ' + (statusMap[json.status] || 'Failed (' + json.status + ')') + ' | Msg: ' + (json.msg
|| ''));"
        ],
        "type": "text/javascript"
      }
    }
  ]
}
```

7.4 Log Management

getlog - Get Historical Logs (Paginated)

POST `{{base_url}}/api`

```
Get historical attendance logs with pagination. Filter by user ID and date range (YYYY-MM-DD). Max 50 records/page.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getlog",
  "password": "{{password}}",
  "index": 0,
  "enrollid": "{{enrollid}}",
  "from": "2024-06-01",
```

```
  "to": "2024-12-31"
}
```

Examples and scripts

```
{
  "event": [
    {
      "listen": "test",
      "script": {
        "exec": [
          "let json = pm.response.json();",
          "pm.test('Total logs: ' + json.count + ' | This page: ' + json.from + '-' + json.to);"
        ],
        "type": "text/javascript"
      }
    }
  ]
}
```

getrtlog - Get Real-Time Logs

POST {{base_url}}/api

```
Get real-time logs (consumed on read). Up to 20 records/page. Logs are removed from queue after reading.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getrtlog",
  "password": "{{password}}",
  "index": 0
}
```

Examples and scripts

```
{
  "event": [
    {
      "listen": "test",
      "script": {
        "exec": [
          "let json = pm.response.json();",
          "pm.test('RT queue count: ' + json.count + ' | Returned: ' + json.from + '-' + json.to);"
```

```
        ],
        "type": "text/javascript"
      }
    }
  ]
}
```

setlogs - Mark Logs as Synced

POST {{base_url}}/api

```
Mark log records as synced/processed.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "setlogs",
  "password": "{{password}}",
  "records": [
    {
      "id": "1001",
      "event": 0,
      "mode": 1,
      "time": "2024-06-01 09:00:00"
    }
  ]
}
```

cleanlog - Clear All Logs [!]

POST {{base_url}}/api

```
[!] DESTRUCTIVE: Removes all historical attendance logs.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
```

```
  ]
}
```

Request body

```
{
  "cmd": "cleanlog",
  "password": "{{password}}"
}
```

cleanlogphoto - Clear All Log Photos

POST `{{base_url}}/api`

```
Clear all attendance log snapshot photos.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "cleanlogphoto",
  "password": "{{password}}"
}
```

7.5 System Management

gettime - Get Device Time

POST `{{base_url}}/api`

```
Get current date/time from the device.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
```

```
    }
  ]
}
```

Request body

```
{
  "cmd": "gettime",
  "password": "{{password}}"
}
```

Examples and scripts

```
{
  "event": [
    {
      "listen": "test",
      "script": {
        "exec": [
          "pm.test('Device time: ' + pm.response.json().time);"
        ],
        "type": "text/javascript"
      }
    }
  ]
}
```

settime - Set Device Time (String Format)

POST `{{base_url}}/api`

```
Sync device time using YYYY-MM-DD HH:MM:SS format.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "settime",
  "password": "{{password}}",
  "cloudtime": "2024-06-01 10:00:00"
}
```

settime - Set Device Time (Unix Timestamp)

POST `{{base_url}}/api`

```
Sync device time using Unix timestamp in milliseconds.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "settime",
  "password": "{{password}}",
  "nowtime": 1717200000
}
```

reboot - Reboot Device [!]

POST `{{base_url}}/api`

```
[!] No response - device reboots immediately. HTTP connection will be dropped.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "reboot",
  "password": "{{password}}"
}
```

disabledevice - Pause Recognition

POST {{base_url}}/api

Disable face/fingerprint/palm/card recognition on the device.

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "disabledevice",
  "password": "{{password}}"
}
```

enabledevice - Resume Recognition

POST {{base_url}}/api

Re-enable face/fingerprint/palm/card recognition on the device.

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "enabledevice",
  "password": "{{password}}"
}
```

initsys - Full System Init [!][!]

POST {{base_url}}/api

```
[!][!] DESTRUCTIVE: Clears ALL users, logs, settings (language preserved), bell schedules, post data, attendance settings, lock data. Backs up WiFi config.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "initsys",
  "password": "{{password}}"
}
```

initmenu - Reset Menu Settings

POST `{{base_url}}/api`

```
Reset menu/settings to default. Does NOT clear users or logs.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "initmenu",
  "password": "{{password}}"
}
```

cleandatebase - Clear Database [!]

POST `{{base_url}}/api`

```
[!] Clears database and reboots device.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "cleandatebase",
  "password": "{{password}}"
}
```

setscreensaver - Set Screensaver

POST `{{base_url}}/api`

```
Configure screensaver image. Set remove=true to delete specific, removeall=true to clear all.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "setscreensaver",
  "password": "{{password}}",
  "filename": "logo.jpg",
  "photo": "<Base64 image>",
  "duration": 60,
  "remove": false,
  "removeall": false
}
```

upgrade - OTA Firmware Upgrade (Chunked)

POST `{{base_url}}/api`

```
Upload firmware in chunks. Send repeatedly incrementing index until all chunks uploaded. MD5 verified on last
```

chunk.

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "upgrade",
  "password": "{{password}}",
  "filename": "ai230_product.zip",
  "md5": "d41d8cd98f00b204e9800998ecf8427e",
  "allcount": 100,
  "index": 0,
  "count": 50,
  "record": "<Base64 firmware chunk>"
}
```

setvoice - Configure Voice Prompt

POST `{{base_url}}/api`

Set voice prompt audio (WAV, max 512KB). Set play=true to test immediately.

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "setvoice",
  "password": "{{password}}",
  "index": 0,
  "voice": "<Base64 WAV audio>",
  "play": true
}
```

keypad - Activate Keypad Mode

POST `{{base_url}}/api`

```
Activate/deactivate keypad mode on device screen.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

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

7.6 Door Access Control

opendoor - Remote Door Open

POST `{{base_url}}/api`

```
Remote door open command. Optional: enrollid (for Wiegand), floor (elevator), msg (screen display 3s).
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "opendoor",
```

```
  "password": "{{password}}",
  "enrollid": "{{enrollid}}",
  "floor": 3,
  "msg": "Welcome"
}
```

lockctrl - Door Lock State Control

POST {{base_url}}/api

```
Control door lock state: NO/NC mode or restore normal operation.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "lockctrl",
  "password": "{{password}}",
  "fuc": 1
}

/* fuc: 1=Normally Open(NO), 2=Normally Closed(NC), 0=Restore normal */
```

getdoorstatus - Get Door Status

POST {{base_url}}/api

```
Query current door status code.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getdoorstatus",
  "password": "{{password}}"
}
```

setdevlock - Set Door Access Parameters

POST `{{base_url}}/api`

```
Configure door access parameters: delay, sensor, alarm, anti-passback, interlock, Wiegand format, time zones, lock groups, visitor times.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "setdevlock",
  "password": "{{password}}",
  "opendelay": 5,
  "doorsensor": 1,
  "alarmdelay": 30,
  "InputAlarm": 0,
  "antpass": 0,
  "interlock": 0,
  "mutiopen": 0,
  "tryalarm": 5,
  "access_times": 0,
  "punchtimes": 0,
  "tamper": 0,
  "wgformat": 26,
  "wgoutput": 0,
  "cardformat": 0,
  "dayzone": [],
  "weekzone": [],
  "lockgroup": [],
  "nopentime": [],
  "visitortime": []
}
```

getdevlock - Get Door Access Parameters

POST `{{base_url}}/api`

```
Get current door access configuration parameters.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getdevlock",
  "password": "{{password}}"
}
```

getuserlock - Get User Access Settings

POST `{{base_url}}/api`

```
Get access control settings for a specific user (timezone, group, validity period, etc.)
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getuserlock",
  "password": "{{password}}",
  "enrollid": "{{enrollid}}"
}
```

setuserlock - Batch Set User Access

POST `{{base_url}}/api`

```
Batch set user-level access parameters (verify mode, timezone, group, validity period, etc.)
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "setuserlock",
  "password": "{{password}}",
  "count": 2,
  "record": [
    {
      "enrollid": "1001",
      "birthday": "01-15",
      "verifymode": 0,
      "weekzone": 0,
      "group": 0,
      "access_times": 0,
      "starttime": "2024-01-01 00:00:00",
      "endtime": "2025-12-31 23:59:59"
    },
    {
      "enrollid": "1002",
      "birthday": "06-20",
      "verifymode": 1,
      "weekzone": 1,
      "group": 0,
      "access_times": 5,
      "starttime": "2024-01-01 00:00:00",
      "endtime": "2025-12-31 23:59:59"
    }
  ]
}
```

deleteuserlock - Delete User Access Settings

POST `{{base_url}}/api`

```
Delete access control settings for a specific user.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "deleteuserlock",
  "password": "{{password}}",
  "enrollid": "{{enrollid}}"
}
```

cleanuserlock - Clear All User Access Settings

POST `{{base_url}}/api`

```
Clear all user-level access control settings.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "cleanuserlock",
  "password": "{{password}}"
}
```

openallfloors - Open All Floors (Elevator)

POST `{{base_url}}/api`

```
Elevator control: open access to all floors.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "openallfloors",
  "password": "{{password}}"
}
```

closeallfloors - Close All Floors (Elevator)

POST `{{base_url}}/api`

```
Elevator control: close access to all floors.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "closeallfloors",
  "password": "{{password}}"
}
```

openselectfloors - Open Selected Floors

POST `{{base_url}}/api`

```
Elevator control: open specific floors (comma-separated floor numbers).
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "openselectfloors",
  "password": "{{password}}",
  "profile": "1,2,5"
}
```

closeselectfloors - Close Selected Floors

POST `{{base_url}}/api`

```
Elevator control: close specific floors (comma-separated floor numbers).
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "closeselectfloors",
  "password": "{{password}}",
  "profile": "3,4"
}
```

7.7 Shifts and Attendance

getshift - Get Shift Configuration

POST `{{base_url}}/api`

```
Get all shift definitions (up to 8 shifts, each with 3 time sections).
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
```

```
  ]
}
```

Request body

```
{
  "cmd": "getshift",
  "password": "{{password}}"
}
```

setshift - Set Shift Configuration

POST `{{base_url}}/api`

```
Configure shift schedule definitions.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "setshift",
  "password": "{{password}}",
  "shifts": [
    {
      "name": "Shift 1",
      "cutoftime": 10,
      "sections": [
        {
          "start": "09:00",
          "end": "12:00",
          "type": 0
        },
        {
          "start": "13:00",
          "end": "18:00",
          "type": 0
        },
        {
          "start": "",
          "end": "",
          "type": 0
        }
      ]
    }
  ]
}
```

getbelltime - Get Bell Schedule

POST {{base_url}}/api

Get bell ringing schedule.

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getbelltime",
  "password": "{{password}}"
}
```

setbelltime - Set Bell Schedule

POST {{base_url}}/api

Set bell ringing schedule.

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "setbelltime",
  "password": "{{password}}",
  "belltimes": [
    {
      "time": "09:00",
      "flag": 1
    },
    {
      "time": "12:00",
      "flag": 0
```

```
    },
    {
      "time": "13:00",
      "flag": 1
    },
    {
      "time": "18:00",
      "flag": 0
    }
  ]
}
```

getholiday - Get Holiday Settings

POST `{{base_url}}/api`

```
Get holiday definitions (up to 30 holidays with shift/dayzone override).
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getholiday",
  "password": "{{password}}"
}
```

setholiday - Set Holiday Settings

POST `{{base_url}}/api`

```
Configure holiday schedule with shift/dayzone overrides.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "setholiday",
  "password": "{{password}}",
  "holidays": [
    {
      "name": "New Year",
      "startday": "01-01",
      "endday": "01-03",
      "shift": 0,
      "dayzone": 0
    }
  ]
}
```

7.8 Reports

getreport - Get Attendance Report Data

POST `{{base_url}}/api`

```
Get attendance report data filtered by user and/or date range.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getreport",
  "password": "{{password}}",
  "enrollid": "{{enrollid}}",
  "from": "2024-01-01",
  "to": "2024-12-31"
}
```

generatereport - Generate Report Files

POST `{{base_url}}/api`

```
Trigger report generation. Download results via /download/report and /download/rawrecord after completion.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "generatereport",
  "password": "{{password}}",
  "enrollid": "{{enrollid}}",
  "from": "2024-01-01",
  "to": "2024-12-31"
}
```

checkreportprogress - Check Report Progress

POST `{{base_url}}/api`

```
Check report generation progress percentage and status.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "checkreportprogress",
  "password": "{{password}}"
}
```

Examples and scripts

```
{
  "event": [
    {
      "listen": "test",
      "script": {
        "exec": [
          "let json = pm.response.json();",
```

```
          "pm.test('Progress: ' + json.progress + '% | Status: ' + json.status + ' | Finished: ' + json.finished);"
        ],
        "type": "text/javascript"
      }
    }
  ]
}
```

7.9 Questionnaire and Notices

setquestionnaire - Configure Questionnaire

POST `{{base_url}}/api`

```
Configure on-device questionnaire (health declaration, survey, etc.).
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "setquestionnaire",
  "password": "{{password}}",
  "title": "Health Declaration",
  "voice": "Please complete health declaration",
  "errmsg": "Declaration failed",
  "radio": true,
  "optionflag": 3,
  "items": [
    "Question 1",
    "Question 2"
  ],
  "schedules": [
    "08:00-09:00",
    "12:00-13:00"
  ],
  "usequestion": true,
  "useschedule": false,
  "card": 0,
  "switch_card": 0,
  "enrollid": "{{enrollid}}",
  "open": true
}
```

getquestionnaire - Get Questionnaire Config

POST {{base_url}}/api

```
Get current questionnaire configuration.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getquestionnaire",
  "password": "{{password}}"
}
```

cleanquestionnaire - Clear Questionnaire

POST {{base_url}}/api

```
Clear all questionnaire settings.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "cleanquestionnaire",
  "password": "{{password}}"
}
```

setcompanyname - Set Company Name

POST {{base_url}}/api

```
Set company name displayed on device (max 21 characters).
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "setcompanyname",
  "password": "{{password}}",
  "name": "ACME Corporation"
}
```

getcompanyname - Get Company Name

POST `{{base_url}}/api`

```
Get currently configured company name.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getcompanyname",
  "password": "{{password}}"
}
```

7.10 Video Intercom (WebRTC)

enablewebrtc - Initiate Video Call

POST `{{base_url}}/api`

```
Initiate WebRTC video call (intercom/monitor mode). 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.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "enablewebrtc",
  "password": "{{password}}",
  "intercom": true,
  "monitor": true,
  "inputmode": 0,
  "audioonly": false,
  "waitseconds": 30,
  "stunserver": "stun:stun.l.google.com:19302",
  "turnserver": "",
  "turnusername": "",
  "turnpassword": ""
}
```

getwebrtc - Get WebRTC Signaling Status

POST `{{base_url}}/api`

```
Get current WebRTC signaling configuration status.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
```

```
  ]
}
```

Request body

```
{
  "cmd": "getwebrtc",
  "password": "{{password}}"
}
```

checklive - Check Live Stream / ICE

POST `{{base_url}}/api`

```
Check live stream status. Returns ICE candidate, callcancel notice, or null when idle. Non-standard response format.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "checklive",
  "password": "{{password}}"
}
```

answer - SDP Answer

POST `{{base_url}}/api`

```
Send SDP answer to device for WebRTC handshake.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "answer",
  "password": "{{password}}",
  "sdp": "<SDP answer string>"
}
```

icecandidate - ICE Candidate Exchange

POST `{{base_url}}/api`

```
Send ICE candidate to device during WebRTC negotiation.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "icecandidate",
  "password": "{{password}}",
  "candidate": "<ICE candidate string>"
}
```

callcancel - Cancel Call

POST `{{base_url}}/api`

```
Cancel active WebRTC video call.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "callcancel",
  "password": "{{password}}"
}
```

monitorcall - Monitor Mode Call

POST `{{base_url}}/api`

```
Start monitor mode video call. Returns SDP offer if successful.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "monitorcall",
  "password": "{{password}}"
}
```

7.11 Biometric Verification

verify - Remote Biometric Verification

POST `{{base_url}}/api`

```
Remote verification using face image, fingerprint templates, card, or password. Provide one or more verification
factors.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "verify",
  "password": "{{password}}",
  "enrollid": "{{enrollid}}",
  "card": "12345678",
  "pwd": 123456,
  "name": "John Doe",
  "face": "<Base64 face JPEG>",
  "finger": "<Hex fingerprint template>",
  "finger2": "<Hex second fingerprint template>",
  "profile": "default"
}
```

biomatch - Biometric Match (OSDP Test)

POST `{{base_url}}/api`

```
OSDP test: trigger biometric match. Requires config_OSDP_TEST setting.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "biomatch",
  "password": "{{password}}"
}
```

bioread - Read Raw Bio Data (OSDP Test)

POST `{{base_url}}/api`

```
OSDP test: read raw biometric data. Requires config_OSDP_TEST setting.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
```

```
  ]
}
```

Request body

```
{
  "cmd": "bioread",
  "password": "{{password}}"
}
```

7.12 File Operations

getdir - List Directory

POST `{{base_url}}/api`

```
List files and subdirectories at given path.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getdir",
  "password": "{{password}}",
  "path": "/data"
}
```

getfile - Get File Content (Base64)

POST `{{base_url}}/api`

```
Read file content returned as Base64 encoded string.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
```

```
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "getfile",
  "password": "{{password}}",
  "path": "/data/config.txt"
}
```

writefile - Write File

POST {{base_url}}/api

```
Write content (Base64 encoded) to a file path on device.
```

Request configuration

```
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Request body

```
{
  "cmd": "writefile",
  "password": "{{password}}",
  "path": "/data/notice.txt",
  "base64": "<Base64 content>"
}
```