API & WebSocket Specification - Hệ Thống Loa Thông Minh (Smart Speaker ESP32)

- **Base HTTP URL:** `http://103.189.208.107:6789`
- **Base WebSocket Endpoint (Control & Device):** `ws://103.189.208.107:6789/ws/control` (hoặc `/ws`)

---

1. Authentication & Authorization

1.1. Đăng nhập (Lấy Token)

Dùng để lấy mã Token JWT trước khi sử dụng các dịch vụ khác (hiệu lực 7 ngày).

- **Đường dẫn:** `/auth/login`
- **Phương thức:** `POST`
- **Headers:** `Content-Type: application/json`
- **Dữ liệu gửi lên (Body):**

```
{
  "username": "admin",
  "password": "admin123"
}
```

- **Kết quả (`200 OK`):** Trả về mã token. Lưu mã này lại để dùng cho Header `Authorization: Bearer <mã_token>` hoặc truyền vào WebSocket `?token=<mã_token>`.

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

1.2. Xác minh (Verify)

Dùng để kiểm tra xem Token còn hiệu lực hay không.

- **Đường dẫn:** `/auth/verify`
- **Phương thức:** `GET`
- **Headers:** `Authorization: Bearer <mã_token>`
- **Kết quả (`200 OK`):**

```
{
  "valid": true
}
```

---

2. Device Provisioning & Onboarding

2.1. Đăng ký thiết bị

- **Phương thức:** `POST`
- **Đường dẫn:** `/api/v1/devices/register`
- **Headers:** `Authorization: Bearer <login_token>`
- **Body (tùy chọn):**

```
{
  "device_id": "dev_001",
  "mac_address": "24:6F:28:AB:CD:EF",
  "model": "SmartSpeaker-ESP32",
  "firmware_version": "1.0.0"
}
```

(Nếu không truyền `device_id`, hệ thống tự sinh `dev_001`, `dev_002`, ...)

- **Response (`200 OK`):**

```
{
  "device_id": "dev_001",
  "device_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "poll_interval_sec": 30
}
```

---

3. Device Control & Commands (WebSocket Realtime)

3.1. Kết nối WebSocket từ thiết bị (ESP32)

```
ws://103.189.208.107:6789/ws/control?device_id=dev_001&token=<device_token>
```

(Hỗ trợ cả endpoint `/ws` và `/ws/control`).

3.2. Cấu trúc lệnh Server gửi cho thiết bị (JSON Body WebSocket)

Các mẫu dưới đây là payload của text message WebSocket gửi tới thiết bị ESP32. Cấu trúc chuẩn: chỉ thay `payload.command` và `payload.params`; phần bao ngoài `type`, `request_id`, `device_id` giữ nguyên.

A. Thay đổi cấu hình Wi-Fi (`wifi.update`)

Thay cấu hình Wi-Fi; có khả năng làm ngắt kết nối WebSocket, nên ACK `received` trước khi đổi mạng.

```
{
  "type": "device.command",
  "request_id": "cmd_1792998000101",
  "device_id": "dev_001",
  "payload": {
    "command": "wifi.update",
    "params": {
      "ssid": "Office_WiFi",
      "password": "<wifi_password>"
    }
  }
}
```

B. Chuyển kênh sang danh sách nhạc phát (`channel.set` - mode playlist)

Chuyển sang chế độ danh sách phát theo playlist.

```
{
  "type": "device.command",
  "request_id": "cmd_1792998000102",
  "device_id": "dev_001",
  "payload": {
    "command": "channel.set",
    "params": {
      "mode": "playlist",
      "playlist_id": "playlist_001"
    }
  }
}
```

C. Chuyển kênh sang phát trực tiếp (`channel.set` - mode live)

Chuyển sang kênh live đang hoạt động.

```
{
  "type": "device.command",
  "request_id": "cmd_1792998000103",
  "device_id": "dev_001",
  "payload": {
    "command": "channel.set",
    "params": {
      "mode": "live",
      "session_id": "live_001",
```

```
      "playback_url": "/live/live_001/stream"
    }
  }
}
```

D. Phát nhạc (`play`)

Phát nội dung / tiếp tục phát.

```
{
  "type": "device.command",
  "request_id": "cmd_1792998000104",
  "device_id": "dev_001",
  "payload": {
    "command": "play",
    "params": {
      "file": "emthuacota.mp3"
    }
  }
}
```

E. Tạm dừng (`pause`)

```
{
  "type": "device.command",
  "request_id": "cmd_1792998000105",
  "device_id": "dev_001",
  "payload": {
    "command": "pause",
    "params": {}
  }
}
```

F. Bài tiếp theo (`next`)

Bài tiếp theo trong playlist / thư mục âm thanh.

```
{
  "type": "device.command",
  "request_id": "cmd_1792998000106",
  "device_id": "dev_001",
  "payload": {
    "command": "next",
    "params": {}
```

```
  }
}
```

G. Bài trước (`previous`)

Bài trước trong playlist / thư mục âm thanh.

```
{
  "type": "device.command",
  "request_id": "cmd_1792998000107",
  "device_id": "dev_001",
  "payload": {
    "command": "previous",
    "params": {}
  }
}
```

H. Dừng phát (`stop`)

Dừng phát tại thiết bị.

```
{
  "type": "device.command",
  "request_id": "cmd_1792998000108",
  "device_id": "dev_001",
  "payload": {
    "command": "stop",
    "params": {}
  }
}
```

I. Đặt âm lượng (`set_volume`)

Âm lượng nguyên từ `0` đến `100`.

```
{
  "type": "device.command",
  "request_id": "cmd_1792998000109",
  "device_id": "dev_001",
  "payload": {
    "command": "set_volume",
    "params": {
      "volume": 65
    }
```

```
    }
}
```

---

4. Telemetry & Data Management

4.1. Kênh phát theo danh sách (Audio Files)

4.1.1. Lấy danh sách nhạc

Trả về danh sách các file MP3 có trong thư mục âm thanh của máy chủ.

- **Đường dẫn:** `/audio`
- **Phương thức:** `GET`
- **Headers:** `Authorization: Bearer <mã_token>`
- **Kết quả (`200 OK`):** Trả về mảng JSON bao gồm Tên file, Dung lượng (bytes), Thời gian cập nhật và URL để Stream.

```
[
  {
    "id": "emthuacota.mp3",
    "name": "emthuacota.mp3",
    "size": 4623820,
    "modified": "2026-10-20T08:00:00.000Z",
    "url": "/audio/stream?file=emthuacota.mp3"
  }
]
```

4.1.2. Stream nhạc (Phát trực tiếp)

Dùng cho tính năng nghe trực tuyến trên ESP32 hoặc trình phát nhạc Web/App.

- **Đường dẫn:** `/audio/stream?file=<tên_file>`
- **Ví dụ:** `/audio/stream?file=emthuacota.mp3`
- **Phương thức:** `GET`
- **Headers:**
  - `Authorization: Bearer <mã_token>` (Hoặc truyền qua query `?token=<mã_token>`)
  - `Range: bytes=0-` *(Tùy chọn: Để tua nhạc hoặc tải nối tiếp sau khi ngắt Mic)*
- **Kết quả (`200 OK` hoặc `206 Partial Content`):** Trả về luồng dữ liệu nhị phân `audio/mpeg`.

4.1.3. Tải nhạc (Download)

Dùng khi loa muốn tải file về lưu vào thẻ nhớ SD để nghe Offline.

- **Đường dẫn:** `/audio/download?file=<tên_file>`
- **Phương thức:** `GET`
- **Headers:** `Authorization: Bearer <mã_token>`

4.1.4. Tải file nhạc mới lên Server (Upload)

Dùng cho Trung tâm tải file `.mp3` (bản tin mới, bài hát mới) lên máy chủ.

- **Đường dẫn:** `/audio/upload`
- **Phương thức:** `POST`
- **Headers:**
  - `Authorization: Bearer <login_token>`
  - `Content-Type: multipart/form-data`
- **Dữ liệu gửi lên (Form-Data):**
  - `file`: *(File `.mp3` cần tải lên, tối đa 50MB)*
  - `filename`: *(Tùy chọn - Tên file muốn lưu trên server)*
- **Kết quả (`201 Created`):**

```
{
  "success": true,
  "message": "Audio file uploaded successfully",
  "file": {
    "id": "ban_tin_sang.mp3",
    "name": "ban_tin_sang.mp3",
    "size": 2450120,
    "modified": "2026-10-20T08:50:00.000Z",
    "url": "/audio/stream?file=ban_tin_sang.mp3"
  }
}
```

4.1.5. Xóa file nhạc khỏi Server

- **Đường dẫn:** `/audio?file=<tên_file>`
- **Ví dụ:** `/audio?file=ban_tin_cu.mp3`
- **Phương thức:** `DELETE`
- **Headers:** `Authorization: Bearer <login_token>`
- **Kết quả (`200 OK`):**

```
{
  "success": true,
  "message": "Audio file 'ban_tin_cu.mp3' deleted successfully",
  "file": "ban_tin_cu.mp3"
}
```

4.2. Kênh phát trực tiếp (Live Stream & Mic Broadcast)

Điều khiển phiên live hoàn toàn qua WebSocket Control, không dùng REST để tạo/bắt đầu/dừng/truy vấn phiên. Dữ liệu âm thanh liên tục là media data-plane riêng qua HTTP binary stream.

- **WebSocket Control Endpoint:** `ws://103.189.208.107:6789/ws/control?role=admin&token=<login_token>`

4.2.1. WebSocket tạo phiên live (`live.session.create`)

- **Client gửi lên WS Control:**

```
{
  "type": "live.session.create",
  "request_id": "live_req_001",
  "payload": {
    "title": "Thông báo trực tiếp",
    "device_ids": [
      "dev_001",
      "dev_002"
    ],
    "audio": {
      "codec": "mp3",
      "sample_rate_hz": 44100,
      "channels": 1,
      "bitrate_kbps": 128
    }
  }
}
```

- **Server phản hồi:**

```
{
  "type": "live.session.created",
  "request_id": "live_req_001",
  "payload": {
    "session_id": "live_001",
    "status": "created",
    "ingest_url": "/live/live_001/audio",
    "playback_url": "/live/live_001/stream"
  }
}
```

4.2.2. Bắt đầu phát và chuyển kênh ngay (`live.session.start`)

- **Client gửi lên WS Control:**

```
{
  "type": "live.session.start",
  "request_id": "live_req_002",
  "payload": {
    "session_id": "live_001"
```

```
  }
}
```

- **Server phản hồi:**

```
{
  "type": "live.session.status",
  "request_id": "live_req_002",
  "payload": {
    "session_id": "live_001",
    "status": "live"
  }
}
```

- **Server tự động gửi lệnh chuyển kênh cho thiết bị qua WebSocket:**

```
{
  "type": "device.command",
  "request_id": "cmd_1792998100000",
  "device_id": "dev_001",
  "payload": {
    "command": "channel.set",
    "params": {
      "mode": "live",
      "session_id": "live_001",
      "playback_url": "/live/live_001/stream"
    }
  }
}
```

4.2.3. Truyền âm thanh trực tiếp (Data Plane)

1. **Media Ingest (Nguồn nói / Trung tâm đẩy âm thanh lên):**
   - **Đường dẫn:** `POST /live/{session_id}/audio`
   - **Headers:** `Content-Type: audio/mpeg`
   - **Body:** Binary bytes liên tục (luồng audio streaming, không phải REST command).
2. **Device Playback (ESP32 hoặc App nhận âm thanh):**
   - **Đường dẫn:** `GET /live/{session_id}/stream`
   - **Headers:** `Authorization: Bearer <token>` (tùy chọn)
   - **Kết quả:** Chunked binary stream `audio/mpeg` thời gian thực.
   - *Khi đang live, độ trễ phụ thuộc encoder, mạng và kích thước buffer của ESP32; không dùng command queue.*

4.2.4. Trạng thái và kết thúc live

Tra cứu trạng thái (`live.session.get`):

- **Client gửi:**

```
{
  "type": "live.session.get",
  "request_id": "live_req_003",
  "payload": {
    "session_id": "live_001"
  }
}
```

- **Server phản hồi:**

```
{
  "type": "live.session.status",
  "request_id": "live_req_003",
  "payload": {
    "session_id": "live_001",
    "status": "live",
    "connected_devices": 2
  }
}
```

Dừng phiên live (`live.session.stop`):

- **Client gửi:**

```
{
  "type": "live.session.stop",
  "request_id": "live_req_004",
  "payload": {
    "session_id": "live_001"
  }
}
```

- **Server phản hồi:**

```
{
  "type": "live.session.status",
  "request_id": "live_req_004",
  "payload": {
    "session_id": "live_001",
    "status": "ended"
  }
}
```

- **Server tự động gửi lệnh hoàn về chế độ playlist cho thiết bị qua WebSocket:**

```
{
  "type": "device.command",
  "request_id": "cmd_1792998200000",
  "device_id": "dev_001",
  "payload": {
    "command": "channel.set",
    "params": {
      "mode": "playlist",
      "playlist_id": "playlist_001"
    }
  }
}
```

---

5. ESP32 Gửi Phản Hồi (ACK) & Báo Cáo Trạng Thái

5.1. Báo nhận lệnh (Command ACK)

Khi ESP32 nhận lệnh (đặc biệt trước khi thay đổi Wi-Fi):

```
{
  "type": "device.command.ack",
  "request_id": "cmd_1792998000101",
  "device_id": "dev_001",
  "status": "received"
}
```

5.2. Báo cáo trạng thái định kỳ hoặc khi đổi trạng thái (`device:status`)

```
{
  "type": "device.status",
  "device_id": "dev_001",
  "data": {
    "mode": "playlist",
    "playback_status": "playing",
    "current_track": "emthuacota.mp3",
    "volume": 65,
    "wifi_status": "connected",
    "ssid": "Office_WiFi",
    "ip_address": "192.168.1.150",
    "rssi": -65,
    "uptime_sec": 3600
  }
}
```

5.3. Báo khi phát hết 1 bài nhạc (`track:finished`)

```
{
  "type": "track.finished",
  "device_id": "dev_001",
  "file": "emthuacota.mp3",
  "auto_next": true
}
```

(Server sẽ tự động gửi lệnh `next` hoặc bài tiếp theo xuống cho thiết bị).