

# Tamagotchi Integration Guide

---

This guide details the steps for the "Tamagotchi" device to interact with the Poliisiauto server.

**Git Repository:** <https://github.com/JalalAmeerzai/Poliisiauto.git>

## Base URLs

- **Production (Render):** <https://poliisiauto.onrender.com>
- **Localhost:** <http://127.0.0.1:8000>

---

## Step 1: Register Device

The device must first register to get an `access_token`. This token is required for all subsequent requests.

**Endpoint:** `POST /api/v1/register`

**cURL:**

```
curl -X POST https://poliisiauto.onrender.com/api/v1/register \  
-H "Content-Type: application/json" \  
-H "Accept: application/json" \  
-d '{  
  "first_name": "Tamagotchi",  
  "last_name": "Device",  
  "email": "device001@example.com",  
  "password": "securepassword",  
  "password_confirmation": "securepassword",  
  "device_name": "Tamagotchi-001"  
}
```

**Sample Response:**

```
{  
  "access_token": "1|Mq8... (long string) ... "  
}
```

Save this `access_token`! You will use it in the `Authorization: Bearer <token>` header.

---

## Step 2: Create a Report (Start Conversation)

To send messages, you first need a "Report" (which acts as a conversation container).

**Endpoint:** `POST /api/v1/reports`

**cURL:**

```
curl -X POST https://poliisiauto.onrender.com/api/v1/reports \  
-H "Authorization: Bearer <YOUR_ACCESS_TOKEN>" \  
-H "Content-Type: application/json" \  
-H "Accept: application/json" \  
-d '{  
  "description": "Bullying incident reported by device",  
  "is_anonymous": 0  
}
```

**Sample Response:**

```
{
  "data": {
    "id": 15,
    "description": "Bullying incident reported by device",
    "report_case_id": 10,
    "reporter_id": 5,
    "handler_id": null,
    "bully_id": null,
    "bullied_id": null,
    "is_anonymous": 0,
    "type": null,
    "opened_at": null,
    "closed_at": null,
    "created_at": "2025-12-13T15:30:00.000000Z",
    "reporter_name": "Tamagotchi Device",
    "bully_name": null,
    "bullied_name": null
  }
}
```

Save the *id* (Report ID) from the response (e.g., 15). You need it to send messages.

---

## Step 3: Send Messages

### A. Send Text Message

**Endpoint:** POST /api/v1/reports/{report\_id}/messages

**cURL:**

```
curl -X POST https://poliisiauto.onrender.com/api/v1/reports/15/messages \
-H "Authorization: Bearer <YOUR_ACCESS_TOKEN>" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{
  "content": "Help, I saw something happening!",
  "is_anonymous": 0,
  "type": "text",
  "lat": 60.1699,
  "lon": 24.9384
}'
```

**Sample Response:**

```
{
  "id": 101
}
```

### B. Send Audio Message

**Endpoint:** POST /api/v1/reports/{report\_id}/messages **Note:** Use multipart/form-data for file uploads.

**cURL:**

```
curl -X POST https://poliisiauto.onrender.com/api/v1/reports/15/messages \
-H "Authorization: Bearer <YOUR_ACCESS_TOKEN>" \
-H "Accept: application/json" \
-F "file=@/path/to/audio.wav" \
-F "type=audio" \
-F "is_anonymous=0" \
-F "lat=60.1699" \
-F "lon=24.9384"
```

#### Sample Response:

```
{
  "id": 102
}
```

## Notification Flow (Server -> Mobile App)

When the device sends a message (Step 3), the server automatically sends a push notification via Firebase Cloud Messaging (FCM).

1. **Server Action:** The server detects the new message and sends a notification to the `teachers` topic.
2. **FCM Payload:** The mobile app receives a data payload like this:

```
{
  "to": "/topics/teachers",
  "notification": {
    "title": "New Message in Case: Case Name",
    "body": "Help, I saw something happening!"
    // OR "Audio message received."
  },
  "data": {
    "message_id": "101"
  }
}
```

## Mobile App Steps

### 1. Register for Notifications

The mobile app (Flutter) must subscribe to the `teachers` topic using the shared credentials.

- **Topic:** `teachers`
- **Method:** Use the Flutter SDK to subscribe the device token to this topic.

### 2. Fetch Message Details

When the notification arrives, the app extracts the `message_id` (e.g., `101`) from the `data` payload and fetches the full details.

**Endpoint:** `GET /api/v1/messages/{id}`

**cURL:**

```
curl -X GET https://poliisiauto.onrender.com/api/v1/messages/101 \
-H "Authorization: Bearer <TEACHER_ACCESS_TOKEN>" \
-H "Accept: application/json"
```

#### Sample Response (Text):

```
{
  "id": 101,
  "content": "Help, I saw something happening!",
  "report_id": 15,
  "author_id": 5,
  "is_anonymous": 0,
  "created_at": "2025-12-13T15:35:00.000000Z",
  "author_name": "Tamagotchi Device",
  "type": "text",
  "lat": 60.1699,
  "lon": 24.9384,
  "file_path": null
}
```

#### Sample Response (Audio):

```
{
  "id": 102,
  "content": "[Audio Message]",
  "report_id": 15,
  "author_id": 5,
  "is_anonymous": 0,
  "created_at": "2025-12-13T15:36:00.000000Z",
  "author_name": "Tamagotchi Device",
  "type": "audio",
  "lat": 60.1699,
  "lon": 24.9384,
  "file_path": "https://poliisiauto.onrender.com/storage/audio/abcdef123456.wav"
}
```

**Action:**

- If `type` is `audio`, use the `file_path` URL to stream/play the audio file.
- If `type` is `text`, display the `content`.