Alt text

Wildlife Watcher User Guide

From wildlife.ai
For Wildlife Watcher project

Download PDF

Trap.NZ real time notifications

This document describes how to set up your Wildlife Watcher camera to send real time notifications to a Trap.NZ server.

What is Trap.nz?

Trap.NZ is a free, online platform and mobile app used in New Zealand for managing predator control projects. It helps individuals and organizations track and manage traps, bait stations, monitoring devices, and biodiversity outcomes related to pest control. The platform is designed to support the vision of a Predator-Free New Zealand by 2050.

Key features and functionalities of Trap.NZ include:

- **Mapping and location**: Users can map trap and bait station locations using GPS and view them on various map layers, including topographic, street, and aerial views.
- **Data recording**: Trap.NZ allows users to record data related to trap visits, catches, bait used, and monitoring results.
- **Offline functionality**: The mobile app can be used offline, syncing data to the website when a connection is available, which is useful in remote areas.
- **Reporting**: Trap.NZ provides tools for generating reports on project progress, catch rates, and other relevant data.
- **Collaboration**: It facilitates collaboration among project coordinators, volunteers, and landowners, allowing them to contribute to and manage projects together.
- **Integration**: The platform can integrate with other tools, such as remote monitoring devices (e.g., Econode and Celium), to streamline data collection and management.
- **Support for various trapping methods**: Trap.NZ supports various trapping methods, including manual trapping, bait stations, and camera traps.

Requirements

Hardware

- A Wildlife Watcher camera
- Access to a LoRaWAN gateway that has internet connection (registered with TTN)

Services

- The Things Network (TTN)
- A Trap.nz account and project (create one or join an exisiting one)
- Domain to store custom webhook.php script
- Basic PHP knowledge

Understanding the Data Flow

Your system will work like this:

1. Wildlife Watcher camera detects animals using a neural network model
2. Device sends data via LoRaWAN to IoT gateway
3. Gateway forwards data to your PHP webhook
4. Webhook processes and sends relevant data to your trap.nz project

---

1 Set Up TTN

Step 1: Register on TTN Console

1. Visit: https://console.thethingsnetwork.org/
2. Register your account

> Select **The Things Stack Sandbox** if prompted.

Step 2: Register Gateway

TTN Register Gateway Documentation

If you're using your own gateway:

1. Navigate to **Gateways** → **Register gateway**
2. Fill in:
   - **Gateway ID**
   - **Gateway EUI** (from gateway device)
   - **Frequency plan** (Asia 920-923MHz (used by TTN Australia))
3. Save and power on the gateway
4. Within "live data", you should be receieving heartbeat messages from the gateway

---

Step 3: Create TTN Application

TTN Register Application Documentation

1. Navigate to **Applications** → **+ Add application**
2. Application ID: Usually specific to a project
3. Choose a name and fill out a breif description
4. Click **Create application**

Step 4: Register Edge Device

TTN Register End Device Documentation

1. Inside your application, click **+ Add end device**
2. Choose:

- **Brand / Model**: Custom or select from known
- **Hardware ver. / Firmware ver. / profile (region)**
- **Frequency plan**: (Asia 920-923MHz (used by TTN Australia))

3. Fill in:

- **JoinEUI (appEUI)**
- **DevEUI** (from WW device)
- **AppKey**

4. Confirm and register
5. Back in your application, you can select your end device and see "live data" for heartbeat statuses from your Wildlife Watcher camera

---

2 Setting up your Webhook

This guide walks you through creating a PHP webhook script that receives data from an IoT gateway (connected to your Wildlife Watcher camera) and forwards heartbeat signals and positive animal detections to a Trap.NZ server.

Step 1: Setting Up Your Webhook Endpoint

Create a PHP file that will serve as your webhook endpoint. This file should:

- Accept POST requests with JSON data
- Validate incoming data format
- Log all requests for debugging
- Return appropriate HTTP status codes
- Handle errors gracefully

Place this file in your web server's document root and ensure it's accessible via URL (e.g., `https://yourdomain.com/webhook.php`).

Step 2: Understanding IoT Gateway Data Structure

Your IoT gateway will typically send JSON data containing:

- **Device identification**: Which edge device sent the data
- **Signal quality metrics**: RSSI, SNR values for connection quality
- **Payload data**: The actual sensor/detection information
- **Timestamps**: When the data was collected and transmitted

The payload will contain either heartbeat data (device status, battery level, SD storage capacity) or detection data (species identified, confidence level, detection timestamp).

Step 3: Communicating with trap.nz API

To send data to trap.nz:

- Use HTTP POST requests with JSON payload
- Include proper authentication headers
- Handle different endpoints for heartbeats vs detections
- Implement retry logic for failed requests
- Log all API responses for troubleshooting

Structure your requests according to trap.nz API requirements, typically including trap ID, timestamp, event type, and relevant metadata.

The Trap.nz API is documented here: https://groundtruth.co.nz/static/drafts/api.html

We utilise one API, which is to send a sensor record:
https://groundtruth.co.nz/static/drafts/api.html#tag/Sensors/operation/getSensorRecords

---

Step 4: Testing Your Implementation

Unit Testing: Test individual components:

- Data parsing and validation
- API communication functions
- Security checks
- Logging functionality

Integration Testing: Test the complete flow:

- Send sample heartbeat data
- Send sample detection data
- Verify trap.nz receives correct information
- Test error conditions