Webhook Integration Guide

Enable Real-Time Data Streaming with EdgeSense.io

EdgeSense's Webhook Integration allows you to send real-time data from your IoT Gateways to external systems, web applications, or third-party services instantly as events occur. This guide outlines how to set up, manage, and secure webhook listeners for seamless data delivery.

---

1. What is a Webhook?

A webhook is an HTTP callback: a way for an app to provide other applications with real-time information. When a subscribed event occurs (e.g., sensor data update, device status change), EdgeSense.io sends an HTTP POST request to the specified endpoint URL with the relevant payload.

---

2. Use Cases

- Push real-time sensor data to cloud dashboards.
- Notify ERP/SCADA systems on threshold breaches.
- Stream events into analytics pipelines (Kafka, Apache Nifi, etc.).
- Trigger automation scripts or alerting mechanisms.

---

3. How to Register a Webhook

Endpoint: POST /api/webhooks/register
Registers a new webhook with your project.

Request Body:

json

CopyEdit

```
{
  "url": "https://your-server.com/webhook-receiver",
```

```
  "events": ["device.data", "device.alert"],
  "secret": "your_shared_secret"
}
```

Parameters:

- url: Your receiving server URL (must support HTTPS).
- events: List of event types to subscribe to.
- secret: Used to sign requests for verification.

Response:

json

CopyEdit

```
{
  "webhookId": "wh_68f93jkdf",
  "status": "registered"
}
```

---

4. Event Payload Example

When an event occurs, EdgeSense.io sends a POST request to your registered webhook URL with the following structure:

json

CopyEdit

```
{
  "event": "device.data",
  "timestamp": "2025-07-24T10:35:17Z",
  "deviceId": "edge-node-321",
  "payload": {
    "temperature": 75.3,
```

```
    "humidity": 42.5,
    "pressure": 101.2
  },
  "signature": "HMAC-SHA256-encoded-string"
}
```

---

5. Verifying Webhook Requests

To ensure the request is from EdgeSense.io, verify the signature header using your shared secret.

Python Example:

```
python
CopyEdit
import hmac
import hashlib

def verify_signature(secret, payload, signature):
    computed = hmac.new(secret.encode(), payload.encode(), hashlib.sha256).hexdigest()
    return hmac.compare_digest(computed, signature)
```

---

6. Supported Events

Event Name	Description
device.data	New sensor reading from device
device.alert	Triggered when thresholds are breached
device.status	Online/offline status of devices
config.update	Configuration changes pushed to devices

---

7. Managing Webhooks

- **List Webhooks**
  GET /api/webhooks
- **Delete Webhook**
  DELETE /api/webhooks/{webhookId}

---

8. Retry Policy

If your endpoint fails to respond with HTTP 2xx, EdgeSense retries up to **5 times** with **exponential backoff**.
After repeated failures, the webhook is **temporarily disabled** until manually re-enabled.

---

9. Best Practices

- Always use **HTTPS** for receiving webhooks.
- Use **HMAC signatures** to verify authenticity.
- Respond quickly (within **3 seconds**) with 200 OK.
- **Queue and process** payloads asynchronously.
- Maintain **endpoint uptime** for reliable delivery.

---

10. Support

For help setting up or debugging your webhook integration, contact us at:

support@edgesense.io