



VERSION 1.0

Sentrax RTLS Hardware Integration Guide



Overview

This document provides a comprehensive guide to integrating Sentrax's Real-Time Location System (RTLS) hardware with partner platforms. Our system offers multiple integration options, including REST APIs, a Positioning Engine with Connector/Docker Images, and MQTT. This guide will walk you through each option, providing the necessary details to facilitate seamless integration.

Contents

Overview	1
1. Introduction	1
2. Integration Options	1
2.1 Cloud Integration (REST API & MQTT)	1
2.1.1 REST API	1
2.1.2 MQTT	1
2.2 On-Premises Deployment	1
2.3 iFrame Embedding	1
3. Getting Started.....	2
3.1 Prerequisites	2
3.2 Authentication	2
4. Hardware Setup	2
4.1 Gateway Installation	2
4.2 Tag Configuration.....	2
5. REST API Integration	3
5.1 Key Endpoints	3
5.2 Sample Requests and Responses	3
6. On-Premises Deployment with Docker Compose	4
6.1 Deployment Steps.....	4
7. MQTT Integration.....	5
7.1 Broker Details.....	5
7.2 Topics and Payloads	5
8. iFrame Integration	6
10. Support and Troubleshooting	7
10.1 Common Issues	7
10.2 Contact Information.....	7

1. Introduction

Sentrax offers cutting-edge RTLS hardware solutions are designed to provide real-time location data with high accuracy and reliability. To ensure flexibility and ease of integration, we offer multiple methods for partner's platforms to interact with our systems. This guide provides comprehensive instructions for integrating Sentrax RTLS with partner platforms via:

- REST APIs (For structured data access)
- MQTT (For real-time streaming)
- On-Premises Deployment (Full control over infrastructure)
- iFrame Embedding (Seamless UI integration)

For support, contact support@sentrax.com.

2. Integration Options

2.1 Cloud Integration (REST API & MQTT)

2.1.1 REST API

Retrieve location data, configure devices, and manage rules.

- Base URL: <https://api.sentrax.com/api/v1/>
- Authentication: API Key (Bearer Token)

2.1.2 MQTT

Lightweight protocol for real-time location updates.

- Broker: app.sentrax.com (Port: 1883)

2.2 On-Premises Deployment

- Run Sentrax's Positioning Engine, APIs, Solix LITE Platform and MQTT broker locally via Docker images.
- Supports hybrid (cloud + on-prem) models.

2.3 iFrame Embedding

Embed Sentrax's Solix Platform into partner dashboards.

3. Getting Started

3.1 Prerequisites

Before starting the integration, ensure you have the following:

- Network access to Sentrax hardware.
- API Key (Contact support@sentrax.com).
- Docker (for on-prem deployment).

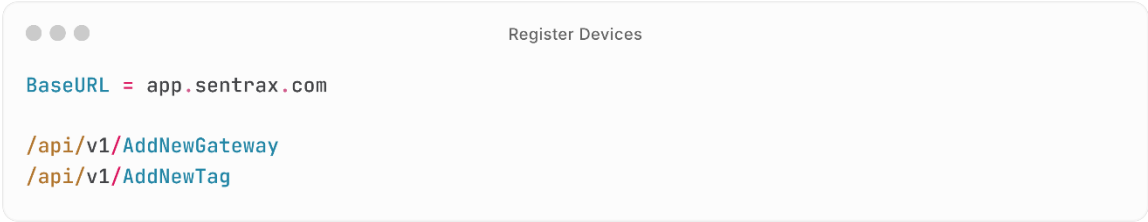
3.2 Authentication

- REST API: Include Authorization: {API_KEY} in headers.
- MQTT: Use credentials provided by Sentrax.

4. Hardware Setup

4.1 Gateway Installation

1. Mount the gateways at optimal locations.
2. Connect to power and network (Ethernet/Wi-Fi/Cellular).
3. **Cloud Deployment:** Verify network connectivity to our cloud services.
4. **On-Premises Deployment:** Ensure connection to the local server (i.e. Local MQTT broker).
5. Register the gateways and tags using the provided APIs.



```
Register Devices

BaseUrl = app.sentrax.com

/api/v1/AddNewGateway
/api/v1/AddNewTag
```

4.2 Tag Configuration

1. Make sure the tag has a battery and is active.
2. Configure tag transmission intervals.
3. Verify connectivity to the nearest gateway.

5. REST API Integration

5.1 Key Endpoints



Key Endpoints

```
GET /api/v1/GetClientAllFloorDevices: Retrieves the floor devices for the client.  
GET /api/v1/GetClientGateways: Retrieves client gateways.  
GET /api/v1/GetClientGatewayHealth: Retrieves the health status of client gateways.  
GET /api/v1/GetClientTags: Retrieves client tags.  
GET /api/v1/GetClientZones: Retrieves client zones along with their building and floor Id.
```

Full API Docs: [Solix API Documentation](#)

5.2 Sample Requests and Responses

Sample request:



Sample Request

```
curl -X GET "https://api.sentrax.com/api/v1/GetClientZones" -H "accept: */*" -H "ClientId:  
1" -H "APIKey: ClientAPIKey"
```

Sample response:



Sample Response

```
{  
  "Message": "Success",  
  "Data": [{  
    "id": 142,  
    "buildingName": "client-building-name",  
    "floorName": "client-floor-name",  
    "name": "client-zone-name",  
    "zoneCorners": "zone-corners-string",  
    "technology": "ble-aoa",  
    "path": "9c726d08-32fb-4779-a57f-f5af9973bff9",  
    "attributes": null,  
    "updatedBy": 0,  
    "createdOn": "2024-11-13 14:53:06",  
    "updatedOn": "2024-11-13 14:53:06"  
  }]  
}
```

6. On-Premises Deployment with Docker Compose

Sentrax provides a fully containerized on-premises solution via Docker Compose, including all required services:

- Local Database (MySQL)
- Local MQTT Broker (Mosquitto)
- Positioning Engine
- REST API
- Web Application

6.1 Deployment Steps

1. Obtain Docker Compose Files

Contact Sentrax Support (support@sentrax.com) to receive:

- `docker-compose.yml` (Main deployment file)
- `.env` (Environment variables)
- Mountable volumes for persistent data

2. Deploy the stack

```
● ● ● On-premises docker compose

# Download the Sentrax package
// File provided via Sentrax's support email
cd sentrax-onprem

# Start all services
docker-compose up -d
```

3. Access Points

```
● ● ● Access points

Service URL Credentials
Web UI http://<host>:4000 Set during installation
API http://<host>:5500 Bearer token
MQTT mqtt://<host>:1883 Set in passwd file
```

7. MQTT Integration

7.1 Broker Details

- **Broker URL:** app.sentrax.com
- **Port:** 1883

7.2 Topics and Payloads

```
MQTT Topics

/Logs/Ping: Device alive ping.
/device/filtered/angles: Retrieves TAG data from Angle of Arrival gateways.
/device/filtered/rssi: Retrieves TAG data from RSSI based gateways.
/tags/locations: Retrives TAG position data from Positioning Engine.
/Notifications: Retrives notifications and alerts.
```

Mqtt topics

```
AoA payload

{
  "mac": "E43819107591",
  "v": "F008V1R6",
  "beacon_data": [
    {
      "beacon_mac": "C4961C69EA7A",
      "RSSI": -70,
      "azimuth": 13,
      "elevation": 65,
      "noise_factor_azimuth": 0,
      "noise_factor_elevation": 0.04,
      "pressure": 100390,
      "humidity": 54,
      "temperature": 17,
      "pdu_data": [
        "0201061AFF4C00021556EF1F00D8A84D5C8B371E20375F2AE707D007D5C4",
        "1EFF410BC4961C69EA7A2110000022A01A960000001137B14EF0F0F0000000"
      ]
    }
  ],
  "timestamp": "2024-11-13T10:52:00Z"
}
```

AoA payload


```

RSSI payload

{
  "mac": "40D63CDB8867",
  "v": "F011V0R3",
  "beacon_data": [
    {
      "beacon_mac": "D88C2DAB297B",
      "rssi": "-75",
      "accelx": 125,
      "accely": 0,
      "accelz": 208
    },
    {
      "beacon_mac": "C35B3BEB556D",
      "rssi": "-78",
      "accelx": 0,
      "accely": 0,
      "accelz": 0
    }
  ]
}
```

RSSI payload

8. iFrame Integration

Embed Sentrax's Solix Platform with device management, configuration and live map into partner dashboards:

```

iFrame Embedded Integration

<iframe
  src="https://app.sentrax.com/embed?client_id=CLIENT123&token=EMBED_TOKEN"
  width="100%"
  height="600px"
  frameborder="0">
</iframe>
```

9. Best Practices

- **Rate Limiting:** Adhere to rate limits to avoid service disruptions.
- **Error Handling:** Implement robust error handling to manage API or MQTT failures.
- **Data Security:** Use secure connections (HTTPS, MQTT over SSL) to protect data.

10. Support and Troubleshooting

10.1 Common Issues

- **Connection Issues:** Ensure network connectivity and correct broker URLs. Check firewall rules.
- **MQTT disconnects:** Enable keepalive (60s).
- **Authentication Failures:** Verify API keys and credentials.
- **Data Inconsistencies:** Check configuration parameters and update intervals.
- **Invalid API Key:** Regenerate keys via Sentrax support.

10.2 Contact Information

For further assistance, please contact our support team at support@sentrax.com or generate a support ticket at www.sentrax.com.