

# Python SDK Guide

Welcome to the official Python SDK documentation for Edgesense.io. This SDK allows you to interact with the Edgesense REST APIs from your Python application with ease, enabling device registration, telemetry upload, configuration changes, and data retrieval.

---

## Installation

Make sure you are using Python 3.7+.

Install the SDK via pip:

```
pip install edgesense-sdk
```

---

## Authentication

Most API endpoints require an API key for authentication. You can generate your API key from the Edgesense Dashboard under:

**Settings → API Keys → Generate Key**

Initialize the client with your API key:

```
from edgesense import EdgeClient
```

```
client = EdgeClient(api_key="YOUR_API_KEY")
```

---

## Basic Usage

### 1. List All Devices

```
devices = client.devices.list()
```

for device in devices:

```
    print(device["name"], device["status"])
```

### 2. Register a New Device

```
new_device = client.devices.register({
    "name": "Boiler-Sensor-1",
    "type": "temperature",
    "location": "Plant Floor A"
})
print("Registered:", new_device["id"])
```

### 3. Send Telemetry Data

```
client.telemetry.send(
    device_id="DEVICE_ID",
    data={
        "temperature": 74.2,
        "humidity": 38,
        "timestamp": "2025-07-24T10:30:00Z"
    }
)
```

### 4. Fetch Telemetry Data

```
data = client.telemetry.fetch(
    device_id="DEVICE_ID",
    start_time="2025-07-24T00:00:00Z",
    end_time="2025-07-24T12:00:00Z"
)
for datapoint in data:
    print(datapoint)
```

---

## SDK Modules

### EdgeClient

Main class for all API interactions.

```
EdgeClient(api_key: str)
```

### **devices**

- `list()` → Returns all registered devices
- `get(device_id)` → Returns details of a specific device
- `register(device_data)` → Registers a new device
- `update(device_id, update_data)` → Updates an existing device
- `delete(device_id)` → Removes a device

### **telemetry**

- `send(device_id, data)` → Sends telemetry data
- `fetch(device_id, start_time, end_time)` → Retrieves historical data

---

## **Error Handling**

The SDK raises specific exceptions for various errors:

```
from edgesense.exceptions import AuthenticationError, ApiError
```

```
try:
```

```
    client.devices.list()
```

```
except AuthenticationError:
```

```
    print("Invalid API key.")
```

```
except ApiError as e:
```

```
    print("API Error:", e.message)
```

---

## **Advanced: Configuration Management**

### **Update Device Configuration**

```
client.devices.update_config("DEVICE_ID", {
```

```
"sampling_rate": "10s",  
"threshold": {  
    "temperature": 80  
}  
})
```

---

## Testing & Mocking

Edgesense SDK is fully compatible with testing frameworks like pytest.

from unittest.mock import patch

```
@patch("edgesense.devices.list")  
def test_device_list(mock_list):  
    mock_list.return_value = [{"name": "TestDevice", "status": "active"}]  
    devices = client.devices.list()  
    assert devices[0]["name"] == "TestDevice"
```

---

## Additional Resources

- [REST API Reference](#)
- [Quick Start Guide](#)
- [Installation & Deployment Manual](#)
- [Support & Help Center](#)

**Contact us for any query at :**

**Email :** [support@edgesense.io](mailto:support@edgesense.io)