📒 SmartEnergy IoT API – Documentation (English Version)

Version: 1.0
Stage: DEV
Base URL:

```
https://luvhyijxnd.execute-api.us-east-1.amazonaws.com/DEV
```

◆ 1. Overview

The **SmartEnergy IoT API** receives sensor events from IoT devices (Shelly, Reyee, CT meters, temperature sensors, and custom hardware) and stores the data in the backend system for energy analytics, prediction models, occupancy detection, and anomaly detection.

All endpoints use:

- **HTTPS**
- **POST**
- **JSON payloads**

The API is designed to be light, fast, and compatible with devices that send small, frequent telemetry packages.

◆ 2. Authentication

Currently, the DEV environment **does not require API keys**.
Production will enforce:

- API Key
- JWT
- IAM Auth (optional)

◆ 3. Endpoints Summary

Endpoint	Method	Description
`/Inventory`	POST	Registers or updates device inventory data
`/IoTDevice`	POST	General endpoint for IoT events
`/lux`	POST	Receives lux/light sensor readings
`/presence`	POST	Receives presence/occupancy sensor events
`/temperature`	POST	Receives temperature sensor data
`/*`	OPTIONS	CORS preflight support

◆ 4. Endpoint Details

✓ 4.1 POST /Inventory

URL:

```
https://luvhyijxnd.execute-api.us-east-1.amazonaws.com/DEV/Inventory
```

Purpose

Stores or updates device metadata (location, type, battery, firmware, owner, etc.).

Sample Request

```
{
  "device_id": "shelly01",
  "device_type": "power_meter",
  "location": "Living Room",
  "firmware": "1.0.8",
  "description": "Shelly EM power monitor"
}
```

Sample Response

```
{
  "ok": true,
  "message": "Inventory record saved successfully"
}
```

✓ 4.2 POST /IoTDevice

URL:

```
https://luvhyijxnd.execute-api.us-east-1.amazonaws.com/DEV/IoTDevice
```

Purpose

Generic endpoint to receive any IoT device event not covered by specific endpoints.

Sample Request

```
{
  "device_id": "iot_002",
  "timestamp": "2025-09-17 12:34:56",
  "event_type": "generic_event",
  "payload": {
    "value": 42,
    "status": "ok"
  }
}
```

Sample Response

```
{
  "ok": true,
  "message": "IoT event processed"
}
```

✓ 4.3 POST /lux

URL:

```
https://luvhyijxnd.execute-api.us-east-1.amazonaws.com/DEV/lux
```

Purpose

Receives **light intensity** (Lux) telemetry for occupancy prediction, automation rules, and dashboards.

Sample Request

```
{
  "device_id": "lux01",
  "ts": "2025-09-17 12:34:56",
  "lux": 300.5
}
```

Sample Response

```
{
  "ok": true,
  "message": "Lux data saved"
}
```

✅ 4.4 POST /presence

URL:

`https://luvhyijxnd.execute-api.us-east-1.amazonaws.com/DEV/presence`

Purpose

Receives presence/occupancy sensor events (PIR, radar, mmWave, smart plugs usage pattern).

Sample Request

```
{
  "device_id": "presence07",
  "ts": "2025-09-17 12:34:56",
  "presence": true
}
```

Sample Response

```
{
  "ok": true,
  "message": "Presence event recorded"
}
```

✅ 4.5 POST /temperature

URL:

`https://luvhyijxnd.execute-api.us-east-1.amazonaws.com/DEV/temperature`

Purpose

Receives temperature telemetry used for heating optimization and energy anomaly detection.

Sample Request

```
{
  "device_id": "temp99",
  "ts": "2025-09-17 12:34:56",
  "temperature_c": 21.8,
  "humidity": 45.6
}
```

Sample Response

```
{
  "ok": true,
  "message": "Temperature data stored"
}
```

◆ 5. Common Error Responses

400 – Bad Request

When required fields are missing.

```
{
  "ok": false,
  "error": "Missing device_id or invalid payload"
}
```

500 – Internal Server Error

Lambda or database failure.

```
{
  "ok": false,
  "error": "Internal server error"
}
```

◆ 6. CORS Information

All endpoints include automatic support for:

```
OPTIONS /
OPTIONS /Inventory
OPTIONS /IoTDevice
OPTIONS /lux
OPTIONS /presence
OPTIONS /temperature
```

CORS headers returned:

```
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, OPTIONS
```

```
Access-Control-Allow-Headers: Content-Type
```

◆ 7. Data Model Requirements

All payloads **must** include:

Field	Type	Required	Notes
`device_id`	string	yes	Unique ID of the IoT device
`ts`	string	recommended	ISO or MySQL timestamp
Sensor Value	number/bool	yes	Depends on endpoint

◆ 8. Logging

API Gateway logs → CloudWatch Logs
Lambda logs → CloudWatch Logs

Each request logs:

- incoming payload
- status
- timestamp
- device_id

◆ 9. Use Cases

- Smart plugs energy consumption
- Room occupancy detection
- Solar energy integration
- HVAC monitoring
- Home/business automation
- Energy anomaly detection
- Predictive maintenance

◆ 10. Contact

Developer: Team # 2
Project: SmartEnergy IoT System
Semester: Fall 2025 – Capstone Full Stack AI Project