Modbus RTU System Monitoring API Documentation

Version: 1.0

Date: December 15, 2025

Base URL: http://localhost:8080/api

Protocol: HTTP/HTTPS

Content-Type: application/json

1. Overview

The Modbus RTU System Monitoring API provides RESTful endpoints for scheduling, managing, and monitoring Modbus RTU jobs that collect system metrics (CPU, RAM, Disk usage) from remote Modbus slaves.

2. Authentication

Currently, the API does not require authentication.

2. API Endpoints

3.1 Create a Job

Schedule a new monitoring job to periodically collect system metrics from a Modbus slave.

Endpoint: POST /api/jobs

Request Headers:
Content-Type: application/json
Request Body:

```
{
  "targetIp": "string",        // IP address or hostname of Modbus slave
  "cronExpression": "string"   // CRON expression for scheduling
}
```

CRON Expression Examples:

- * * * * * - Every minute
- 0 * * * * - Every hour
- 0 0 * * * - Daily at midnight
- 0 */6 * * * - Every 6 hours
- 0 9 * * 1-5 - Every weekday at 9 AM

Response: 200 OK

```
{
  "id": "76221913-32ea-4e74-8289-0285677271ca",
  "targetIp": "192.168.1.100",
  "cronExpression": "* * * * *",
  "status": "RUNNING",
  "createdAt": "2025-12-15T10:30:00"
}
```

Error Responses:

- **400 Bad Request:** {"error": "Invalid CRON expression"}
- **500 Internal Server Error:** {"error": "Failed to schedule job"}

3.2 Get Job Details with Pagination

Retrieve detailed information about a specific job, including its execution history with pagination support.

Endpoint: GET /api/jobs/{jobId}

Path Parameters:

- jobId (string, required) - Unique job identifier

Query Parameters:

- page (integer, optional, default: 0) - Zero-based page number
- size (integer, optional, default: 20, max: 100) - Number of executions per page

Response: 200 OK

```
{
  "jobId": "76221913-32ea-4e74-8289-0285677271ca",
  "status": "RUNNING",
  "executions": [
    {
      "executionId": "exec-123",
      "executionTime": "2025-12-15T10:31:00",
      "status": "COMPLETED",
      "telemetry": {
        "cpu": 45.50,
        "ram": 62.80,
        "disk": 78.50
      }
    },
    {
      "executionId": "exec-124",
      "executionTime": "2025-12-15T10:32:00",
      "status": "ERROR_TIMEOUT",
      "telemetry": null
    }
  ],
  "pagination": {
    "currentPage": 0,
    "pageSize": 20,
    "totalElements": 156,
    "totalPages": 8,
    "first": true,
    "last": false
  }
}
```

Execution Status Values:

- PENDING: Execution is scheduled but not yet started
- COMPLETED: Execution finished successfully with telemetry data
- ERROR_APP: Application-level error
- ERROR_TCP: TCP connection error
- ERROR_TIMEOUT: Request timeout
- ERROR_MODBUS: Modbus protocol error

3.3 List All Jobs

Retrieve a list of all monitoring jobs.

Endpoint: GET /api/jobs

Response: 200 OK

```
[
  {
    "id": "76221913-32ea-4e74-8289-0285677271ca",
    "targetIp": "192.168.1.100",
    "cronExpression": "* * * * *",
    "status": "RUNNING",
    "createdAt": "2025-12-15T10:30:00"
  }
]
```

3.4 Update/Restart Job

Update an existing job's configuration (target IP and/or CRON expression) and restart it.

Endpoint: PATCH /api/jobs/{jobId}

Path Parameters:

- jobId (string, required) - Unique job identifier

Request Body:

```
{
  "targetIp": "string",        // New target IP (optional)
  "cronExpression": "string"   // New CRON expression (optional)
}
```

Response: 200 OK

```
{
  "id": "76221913-32ea-4e74-8289-0285677271ca",
  "targetIp": "192.168.1.102",
  "cronExpression": "0 * * * *",
  "status": "RUNNING",
  "createdAt": "2025-12-15T10:30:00"
}
```

3.5 Stop Job

Stop a running job

Endpoint: DELETE /api/jobs/{jobId}

Path Parameters:

- jobId (string, required) - Unique job identifier

Response: 200 OK

```
{
  "message": "Job stopped successfully"
}
```

4. Data Models

Job Object

Represents a scheduled monitoring job.

Field	Type	Description
id	UUID	Unique job identifier
targetIp	string	IP address or hostname of Modbus slave
cronExpression	string	CRON scheduling expression
status	string	"RUNNING" or "STOPPED"
createdAt	string	ISO 8601 timestamp

Telemetry Object

System metrics collected from Modbus slave.

Field	Type	Description
cpu	number	CPU usage percentage (0.00-100.00)
ram	number	RAM usage percentage (0.00-100.00)

disk	number	Disk usage percentage (0.00-100.00)

5. Pagination

The API supports pagination for job execution history.

Parameters:

- **page:** Zero-based page index (default: 0)
- **size:** Number of items per page (default: 20, max: 100)

Navigation Example:

- First page: /api/jobs/{id}?page=0&size=20
- Next page: /api/jobs/{id}?page=1&size=20

6. Error Handling

All errors follow a consistent JSON format:

```
{
  "timestamp": "2025-12-15T10:30:00",
  "status": 404,
  "error": "Not Found",
  "message": "Job not found with id: ...",
  "path": "/api/jobs/..."
}
```

HTTP Status Codes:

- 200 OK: Successful operation
- 400 Bad Request: Invalid parameters or body
- 404 Not Found: Resource (job) not found
- 500 Internal Server Error: Server-side error

7. Operational Details

Health Check

Monitor API health status.
Endpoint: GET /actuator/health

Modbus Protocol Details

- **Port:** 5000 (configurable via Modbus slave)
- **Protocol:** Modbus RTU over TCP

- **Function Code:** 0x03 (Read Holding Registers) or 0x04 (Read Input Registers)
- **Byte Order:** Big-endian (network byte order)
- **Error Detection:** CRC-16

Register Mapping:

Register	Metric	Data Type	Scale	Range
0x04	CPU Usage	uint16	×100	0.00% - 100.00%
0x06	RAM Usage	uint16	×100	0.00% - 100.00%
0x08	Disk Usage	uint16	×100	0.00% - 100.00%