IOT DEVICE API

API for accessing IIoT device information.

OVERVIEW

This document describes the API that students will implement for the final project in SER486. Request will be made via HTTP over the local area network. Responses should be given as a JSON payload encoded in the body section of the HTTP message response.

More detail about the HTTP protocol can be found in IETF RFC 2626 and the class lecture notes. More information about JSON can be found in IETF RFC 8259.

IIOT DEVICE ENDPOINTS

Endpoints for this IIoT device are resources that can be accessed in order to receive information and/or cause change to the IIoT device state. The endpoints are summarized in the following table.

Endpoint	Description	Operations Supported
\device	A resource that represents the entire IIoT device	GET, PUT
\device\config	A resource that represents the IIoT device configuration	PUT
\device\log	A resource that represents the IIoT device log	DELETE

JSON OBJECT MODELS

JSON is a text based data exchange format that supports specification of data in key value pairs of the style:

"key":value

Values must be a string (enclosed in double quote characters), a number, an object, array, or boolean. Objects are identified by a beginning curly brace character, and conclude with a terminating curly brace character. Objects themselves may contain one or more key/value pairs,

or be completely empty. If multiple key/value pairs are present, they are separated by the comma character as shown below.

{“key1”:20,”key2”:”text”,”key3”:true}

Arrays are specified by an opening square bracket, followed by a closing square bracket. Arrays contain zero or more values which may also be objects, or additional arrays.

The primary object used for SER486 is the Device object. It is returned by GET requests from /device. The device object represents the entire state of the device and is composed of several key/value pairs including an object representing the VPD information and an array of log entry objects. You will be responsible for returning data in the recognizable JSON format for credit in the final project. Definitions of the objects with examples are shown below.

VPD OBJECT DEFINITION

Key	Value Type	Example
“model”	String	“XYZ-2000”
“manufacturer”	String	“Dougtronic”
“serial_number”	String	“asurite”
“manufacture_date”	String	“01/01/2019”
“mac_address”	String	“44:4F:55:53:41:4E”
“country_code”	String	“USA”

Example:

{"model":"Sandy","manufacturer":"Douglas","serial_number":"_UNASSIGNED","manufacture_date":"01/01/2000","mac_address":"44:4F:55:53:41:4E","country_code":"USA"}

LOGENTRY OBJECT DEFINITION

Key	Value Type	Example
“timestamp”	String	"01/01/2000 00:00:00"
“eventnum”	Integer	0

Example:

{"timestamp":"01/01/2000 00:00:07","event":2}

DEVICE OBJECT DEFINITION

Key	Value Type	Example
“vpd”	VPD object	{“model”:“XYZ-2000”, “manufacturer”:“Dougtronic”, “serial_number”:“asurite”,

		“manufacture_date”:“01/01/2019”, “mac_address”:“44:4F:55:53:41:4E”, “country_code”:“USA”}
“tcrit_hi”	Integer	1023
“twarn_hi”	Integer	1022
“tcrit_lo”	Integer	0
“twarn_lo”	Integer	1
“temperature”	Integer	88
“state”		"CRIT_HI”, “WARN_HI", “NORMAL”, “WARN_LO”, “CRIT_LO” depending on the current temperature reading (no hysteresis)
“log”	LogEntry[]	[{“timestamp”:"04/01/2018 00:01:01",“event”:1}, {“timestamp”:"01/01/2018 00:01:00",“event”:0}]

Example:

{"vpd":{"model":"Sandy","manufacturer":"Douglas","serial_number":"asurite","manufacture_date":"01/01/2000","mac_address":"44:4F:55:53:41:4E","country_code":"USA"},"tcrit_hi":1023,"twarn_hi":1022,"tcrit_lo":0,"twarn_lo":1,"temperature":75,"state":"NORMAL","log":[{"timestamp":"01/01/2000 00:00:00","event":3},{"timestamp":"01/01/2000 00:00:00","event":4},{"timestamp":"01/01/2000 00:00:00","event":0},{"timestamp":"01/01/2000 00:00:07","event":2},{"timestamp":"01/01/2000 00:00:01","event":3},{"timestamp":"01/01/2000 00:00:00","event":4},{"timestamp":"01/01/2000 00:00:00","event":0}]}

API CALLS

This document section describes API calls that must be supported by the student.

Note: Example CURL usage should be typed into the Linux terminal. DO NOT COPY-PASTE as some special characters may be inserted in the process, causing incorrect data to be sent to your device.

GET DEVICE STATE

DESCRIPTION:

Returns the complete device state as a Device JSON object.

REQUEST PARAMETERS:

None

EXAMPLE REQUEST FROM LINUX TERMINAL

Request:

curl -X GET 'http://127.0.0.100:8080/device'

The request may also be sent from the Firefox browser as: http://127.0.0.100:8080/device

Response:

{"vpd":{"model":"Sandy","manufacturer":"Douglas","serial_number":"asurite","manufacture_date":"01/01/2000","mac_address":"44:4F:55:53:41:4E","country_code":"USA"},"tcrit_hi":1023,"twarn_hi":1022,"tcrit_lo":0,"twarn_lo":1,"temperature":88,"state":"NORMAL","log":[{"timestamp":"01/01/2000 00:00:00","event":3},{"timestamp":"01/01/2000 00:00:00","event":4},{"timestamp":"01/01/2000 00:00:00","event":0},{"timestamp":"01/01/2000 00:00:07","event":2},{"timestamp":"01/01/2000 00:00:01","event":3},{"timestamp":"01/01/2000 00:00:00","event":4},{"timestamp":"01/01/2000 00:00:00","event":0}]}

SET A TEMPERATURE LIMIT

DESCRIPTION:

Sets a single temperature limit based on the parameter passed. Range checking is performed in order to make sure that high tcrit_hi > twarn_hi > twarn_lo > tcrit_lo. If an invalid parameter is passed, error 400 is returned.

REQUEST PARAMETERS:

Parameter	Type	Description
tcrit_hi	Integer	This parameter will be used to set the new value for the critical high temperature limit of the device. When temperatures exceed this value, the device will send an alarm and update the device's state accordingly. When setting this parameter, it must have a value greater than twarn_hi. This parameter cannot be used in conjunction with other parameters.
twarn_hi	Integer	This parameter will be used to set the new value for the warning high temperature limit of the device. When temperatures exceed this value, the device will send an alarm and update the device's state accordingly. When setting this parameter, it must have a value greater than twarn_lo, but less than tcrit_hi. This parameter cannot be used in conjunction with other parameters.

Parameter	Type	Description
tcrit_lo	Integer	This parameter will be used to set the new value for the warning low temperature limit of the device. When temperatures are below this value, the device will send an alarm and update the device's state accordingly. When setting this parameter, it must have a value greater than tcrit_lo, but less than twarn_hi. This parameter cannot be used in conjunction with other parameters.
twarn_lo	Integer	This parameter will be used to set the new value for the critical low temperature limit of the device. When temperatures are below this value, the device will send an alarm and update the device's state accordingly. When setting this parameter, it must have a value less than twarn_lo. This parameter cannot be used in conjunction with other parameters.

EXAMPLE REQUEST FROM LINUX TERMINAL

```
curl -X PUT 'http://127.0.0.100:8080/device/config?twarn_hi=85'
```

CLEAR ALL LOG ENTRIES

DESCRIPTION:

Clears all log entries from the device's event log.

REQUEST PARAMETERS:

None

EXAMPLE REQUEST FROM LINUX TERMINAL:

```
curl -X DELETE 'http://127.0.0.100:8080/device/log'
```

RESET THE DEVICE

DESCRIPTION:

Force a hardware reset of the device.

REQUEST PARAMETERS:

Parameter	Type	Description
Reset	Boolean	If this parameter is “true”, the device will reset. If the value is “false”, it will do nothing. Other values will generate a 400 error.

EXAMPLE REQUEST FROM LINUX TERMINAL:

```
curl -X PUT 'http://127.0.0.100:8080/device?reset="true"'
```