

Falk V-Class IIoT Smart API Documentation

Introduction

This is the developer documentation for the SMART API. The SMART API is REST based and allows customers to programmatically access information from their Rexnord SMART products.

SMART API Purpose

The SMART API supports programmatic information access for Rexnord customers who own SMART products. The services allow customers to retrieve a product's status and historic information. The API supplies the information in JSON format, so customer applications can easily consume and process the data. This enables customers to use the SMART API to supply historic information for their own data analysis, and status information for tools like HMI displays.

SMART Device Messaging

Each SMART device is identified by a unique identifier. The SMART API refers to this identifier as the *deviceId*. API users can request the list of *deviceId*'s which they are authorized to view. Given that list, users can request information for a specific device by using the *deviceId*.

As Rexnord's SMART devices operate, they produce messages that are sent to a backend cloud repository. SMART devices send two types of messages: *metrics* and *events*.

Metrics messages contain a snapshot of the device's sensor readings at a moment in time. SMART devices send metrics messages at a regular interval, every 15 minutes for example.

Event messages indicate an alert condition has either started or ended for the device. SMART devices send event messages only when an alert condition starts or ends. Event messages contain a *state* element, which can have a value of *latched* or *unlatched*. Messages with a state of *latched* indicate that an alert condition has started. Messages with a state of *unlatched* indicate that an alert condition has ended. For example, when a temperature sensor reading exceeds a threshold an event message is sent with a state of *latched*. When the temperature sensor reading falls back below the threshold, an event message is sent with a state of *unlatched*.

API Overview

API Architecture

The SMART API is a RESTful API, designed to allow customer applications to easily access device data from computers attached to the internet. The API requires authentication to protect against unauthorized access. Response data is supplied in JSON format for easy consumption of the results.

The SMART API is divided into five types of requests illustrated in *Figure 1: SMART API Request Types*.

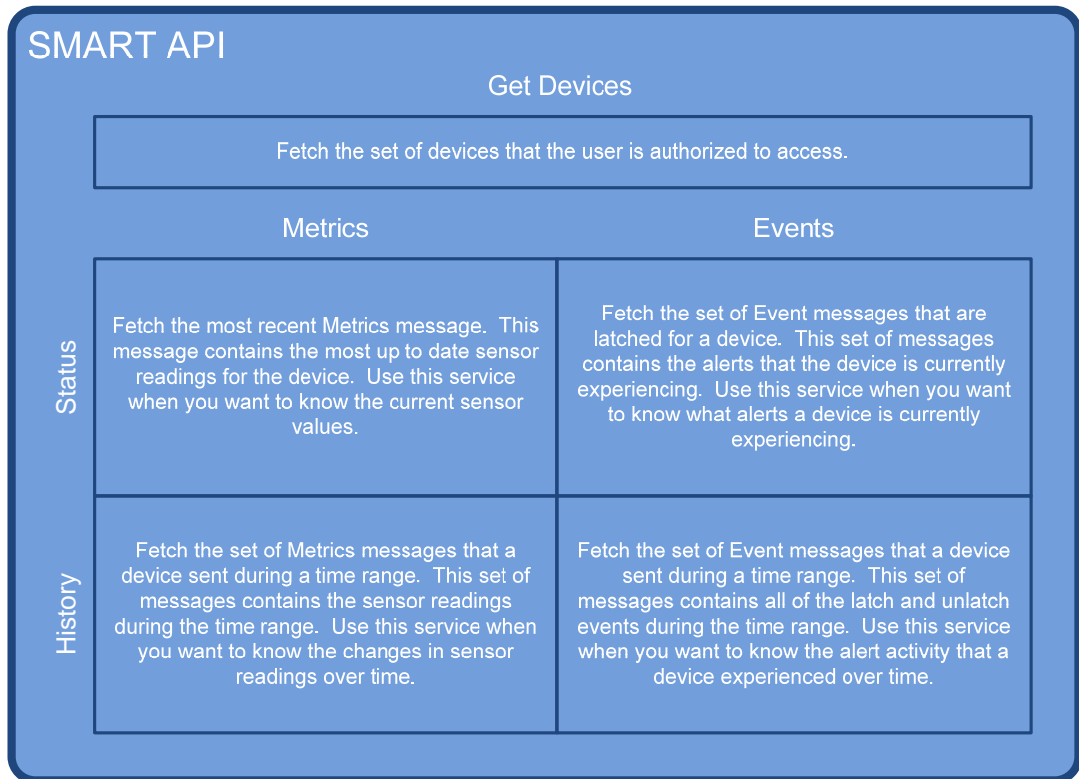


Figure 1: SMART API Request Types

As previously described, SMART devices are identified by a unique identifier called the deviceId, and they send Metrics messages and Event messages. The SMART API provides services to request the set of devices the user can access, as well as the status and the history of Metrics and Event messages for a specified device, identified by a deviceId.

API Security

The API is secured through HTTP basic access authentication. When making a request, the customer provides a user name and password passed via the HTTP Authorization header. The SMART services validate this information, confirming:

- a) That the supplied credentials correctly identify an account that has been granted access to use the SMART API.

- b) For services other than Get Devices, that the authenticated account has authorization to view information for the device specified by deviceId.

Acquiring Credentials

The SMART API uses the Rexnord SMART Portal accounts. If you have a SMART Portal account, your portal credentials can be used to access the SMART API. If you do not have a SMART Portal account you can browser to the portal at: <http://gear.iiotsmart.rexnord.com> and follow the instructions for creating an account.

Acquiring Authorization for a Device

Similar to credentials, the SMART API uses the Rexnord SMART Portal for device authorization. If your account is authorized to view data for a particular device on the portal, then your account can also access the device data via the SMART API. Device authorization is controlled by associates within your company who have Rexnord Vclass portal administrative authorization. If you need authorization to access a specific device, you would request that access from your company's portal administrator. If you do not know your company's portal administrators, send an email to SMART_Support@rexnord.com and request a list of the portal administrators.

API Endpoints

Root URL

The SMART API uses the following root URL:

<https://api.us.apiconnect.ibmcloud.com/rexnord-llc-prod/production/api/iiot-device-apis/>

API Paths

Get Devices

Fetch the set of devices, identified by deviceId, that the authenticated user is authorized to access.

- **URL**
/get_devices
- **Method:**
GET
- **URL Params**
Required:
None
Optional:
None
- **Headers**
Required:
Accept: application/json

Authorization: [HTTP Basic Authentication credentials] (See API Security)

- **Success Response:**

- **Code: 200**
Content: { "devices": ["Simetrics9620001","Simetrics9620002","Simetrics9620007"] }

- **Error Response:**

- **Code: 401 UNAUTHORIZED**
Content: Unauthorized

- **Sample Call:**

Request the set of devices that this account is authorized to access.

```
curl --request GET \  
  --url 'https://api.us.apiconnect.ibmcloud.com /rexnord-llc-prod/production/api/iiot-device-apis/get_devices\  
  --header 'accept: application/json' \  
  --header 'authorization: REPLACE_THIS_VALUE'
```

- **Notes:**

None.

Event History

Fetch the historic event messages for a specified device that occurred within a time range, with support for optionally setting the beginning and ending of that time range.

- **URL**

/event_history

- **Method:**

GET

- **URL Params**

Required:

deviceid=[alphanumeric]

Optional:

startdatetime=[ISO8601 formatted date]

A date, in ISO8601 format, specifying the start of the time range. If not specified, the default is 7 days prior to the current date/time.

enddatetime=[ISO8601 formatted date]

A date, in ISO8601 format, specifying the end of the time range. If not specified, the default is the current date/time.

- **Headers**

Required:

Accept: application/json

Authorization: [HTTP Basic Authentication credentials] (See API Security)

- **Success Response:**

- **Code: 200**
Content: { "events": [] }

- **Error Response:**

- **Code: 401 UNAUTHORIZED**
Content: Unauthorized

OR

- **Code: 400 BAD REQUEST**
Content: { "error": { "statusCode": 400, "name": "Device authorization error", "message": "User not authorized for device" } }

- **Sample Call:**

Request the Event History messages for deviceid=Simetrics0000 within the time range of 2017-04-19T08:00:00-05:00 to 2017-04-22T08:00:00-05:00

```
curl --request GET \  
  --url 'https://api.us.apiconnect.ibmcloud.com/rexnord-llc-prod/production/api/iiot-device-apis/event_history?\  
deviceid=Simetrics0000&startdatetime=2017-04-19T08:00:00-05:00\  
&enddatetime=2017-04-22T08:00:00-05:00' \  
  --header 'accept: application/json' \  
  --header 'authorization: REPLACE_THIS_VALUE'
```

- **Notes:**

Ensure that the query string is URL encoded. In particular, the '+' character in the ISO8601 date/time entries must be properly escaped.

Event Status

Fetch the currently latched event messages for a specified device.

- **URL**

/event_status

- **Method:**

GET

- **URL Params**

Required:

deviceid=[alphanumeric]

- **Headers**

Required:

Accept: application/json

Authorization: [HTTP Basic Authentication credentials] (See API Security)

- **Success Response:**

- **Code: 200**
Content: {"event_status": [{ "Event": { "Name": "ST_START", "Source": "SS_RPM", "Value": "90.5410", "Threshold": "6.0000", "State": "Latched", "Description": "", "Timestamp": "2017-04-13T08:44:41.1456918+00:00" } }] }

- **Error Response:**

- **Code: 401 UNAUTHORIZED**
Content: Unauthorized

OR

- **Code: 400 BAD REQUEST**

Content: { "error": { "statusCode": 400, "name": "Device authorization error", "message": "User not authorized for device" } }

- **Sample Call:**

Request the Event Status for deviceid=Simetrics0000

```
curl --request GET \  
  --url 'https://api.us.apiconnect.ibmcloud.com/rexnord-llc-prod/production/api/iiot-device-apis/event_status?deviceid=Simetrics0000\  
  --header 'accept: application/json' \  
  --header 'authorization: REPLACE_THIS_VALUE'
```

- **Notes:**

None

Metrics History

Fetch the historic metrics messages for a specified device that occurred within a time range, with support for optionally setting the beginning and ending of that time range.

- **URL**

/metrics_history

- **Method:**

GET

- **URL Params**

Required:

deviceid=[alphanumeric]

- **Headers**

Required:

Accept: application/json

Authorization: [HTTP Basic Authentication credentials] (See API Security)

Optional:

startdatetime=[ISO8601 formatted date]

A date, in ISO8601 format, specifying the start of the time range. If not specified, the default is 7 days prior to the current date/time.

enddatetime=[ISO8601 formatted date]

A date, in ISO8601 format, specifying the end of the time range. If not specified, the default is the current date/time.

- **Success Response:**

- **Code: 200**

Content: {"metrics":[{"deviceId":"SimetricsABCDEFGH","timestamp": "2018-01-01T01:11:22.3+00:00","Temp_Oil":75.7,"Motor_Amps": null,"Oil_Coarse_Part particulate": 0,"Oil_Fine_Part particulate": 12.7,"Oil_Oxidation": 0,"Oil_RH": 4.5,"Output_Shaft_RPM": 0,"Temp_Ambient": 72.3,"Temp_Inboard_Bearing": 77.1,"Temp_Outboard_Bearing": 79.2,"Vibe_Input_Shaft": 0.000507915079123538,"Vibe_Output_Shaft": 0}]}

- **Error Response:**

- **Code:** 401 UNAUTHORIZED
Content: Unauthorized
 - OR
 - **Code:** 400 BAD REQUEST
Content: { "error": {"statusCode": 400,"name": "Device authorization error", "message": "User not authorized for device" } }
- **Sample Call:**

Request the Metrics History messages for deviceid=Simetrics0000 within the time range of 2017-04-19T08:00:00-05:00 to 2017-04-22T08:00:00-05:00

```
curl --request GET \  
  --url 'https://api.us.apiconnect.ibmcloud.com /rexnord-llc-prod/production/api/iiot-device-apis/metrics_history?\  
deviceid=Simetrics0000&startdatetime=2017-04-19T08:00:00-05:00\  
enddatetime=2017-04-22T08:00:00-05:00' \  
  --header 'accept: application/json' \  

```

Notes:

Ensure that the query string is URL encoded. In particular, the '+' character in the ISO8601 date/time entries must be properly escaped.

Creating a large time window through explicitly setting the startdatetime and enddatetime can result in the return taking a long period of time.

Metrics Status

Fetch the most recent metrics messages for a specified device.

- **URL**
`/metrics_status`
- **Method:**
GET
- **URL Params**
Required:
`deviceid=[alphanumeric]`
- **Headers**
Required:
Accept: application/json
Authorization: [HTTP Basic Authentication credentials] (See API Security)
- **Success Response:**
 - **Code:** 200
Content: {"metrics_status":{"deviceId":"SimetricsABCDEF","timestamp": "2018-01-01T01:11:22.3+00:00","Temp_Oil":75.7,"Motor_Amps": null,"Oil_Coarse_Part particulate": 0,"Oil_Fine_Part particulate": 12.7,"Oil_Oxidation": 0,"Oil_RH": 4.5,"Output_Shift_RPM": 0,"Temp_Ambient": 72.3,"Temp_Inboard_Bearing": 77.1,"Temp_Outboard_Bearing": 79.2,"Vibe_Input_Shift": 0.000507915079123538,"Vibe_Output_Shift": 0}}
- **Error Response:**
 - **Code:** 401 UNAUTHORIZED
Content: Unauthorized

OR

- **Code:** 400 BAD REQUEST
Content: { "error": { "statusCode": 400, "name": "Device authorization error", "message": "User not authorized for device" } }

- **Sample Call:**

Request the Metrics Status for deviceid=Simetrics0000

```
curl --request GET \
  --url 'https://api.us.apiconnect.ibmcloud.com /rexnord-llc-prod/production/api/iiot-device-apis/metrics_status?deviceid=Simetrics0000\
  --header 'accept: application/json' \
  --header 'authorization: REPLACE_THIS_VALUE'
```

- **Notes:**

None

Additional Support

If you have questions or need additional support, please email SMART_Support@rexnord.com.

Appendix A – Product Message Formats

V-Class Gearbox Message Format

Event Message

V-Class Gearbox Metrics messages use the following format, with annotation:

```
{
  // The _id property is an identifier that is unique to this specific Event
  "_id": "Simetrics9620028 OIL_PARTICULATE_YELLOW Unlatched 2018-01-12T20:08:05.3944333+00:00",
  // The payload property is a container
  "payload": {
    // The Event property is a container for the details of the Event
    "Event": {
      // The Name property is a human readable title for the Event
      "Name": "OIL_PARTICULATE_YELLOW",
      // The Source property indicates the source of the Event message. Valid values are "Bluemix"
      // indicating the message was generated in the cloud analytics backend, and any other name which
      // should be interpreted as the sensor on the edge device that generated the Event
      "Source": "Bluemix",
      // The Value property indicates the value of the sensor or computed value which triggered the
      // Event
      "Value": 8.51317405700684,
      // The Threshold property indicates the threshold value that the sensor passed through to
      // trigger the Event
      "Threshold": 75,
      // The State property indicates the whether the Event is beginning or ending. Valid values are
      // 'Latched', indicating the Event is starting, and "Unlatched" indicating the Event is ending.
      "State": "Unlatched",
    }
  }
}
```



```

// The Description property is a human readable description of the Event. If the 'State' is
"Unlatched", description is typically blank
    "Description": "",
// The Timestamp property is the date/time of the Metrics Event in ISO8601 format
    "Timestamp": "2018-01-12T20:08:05.3944333+00:00",
// The Severity property is an assessment of the criticality of the Event. Valid values are
"Low", "Medium", "High".
    "Severity": "Medium"
  }
},
// The deviceId property is the unique deviceId identifying the device experiencing the Event
"deviceId": "Simetrics9620028"
}

```

Metrics Message

V-Class Gearbox Metrics messages use the following format, with annotation:

```

{
  // The deviceId property is the unique deviceId identifying the requested device
  "deviceId": "Simetrics9620028",
  // The timestamp property is the date/time of the Metrics Event in ISO8601 format
  "timestamp": "2018-01-11T21:06:16.337826+00:00",
  // The Oil_Coarse_Part particulate property is a scalar value representing the coarse oil particulate.
  Higher values indicate more particulate.
  "Oil_Coarse_Part": 5.80988264083862,
  // The Oil_Fine_Part particulate property is a scalar value representing the fine oil particulate.
  Higher values indicate more particulate.
  "Oil_Fine_Part": 7.7589635848999,
  // The Oil_Oxidation property is scalar value representing the high frequency oil impedance.
  This is an indicator of oil health with lower values indicating less 'healthy' oil.
  "Oil_Oxidation": 0,
  // The Oil_RH property is relative humidity of the oil represented in percent.
  "Oil_RH": 1.67003047466278,
  // The Output_Shaft_RPM property is the speed of the low speed shaft in revolutions per
  minute.
  "Output_Shaft_RPM": 0,
  // The Temp_Ambient property is the ambient temperature in degrees Fahrenheit.
  "Temp_Ambient": 88.585302734375,
  // The Temp_Inboard_Bearing property is the inboard bearing temperature in degrees
  Fahrenheit.
  "Temp_Inboard_Bearing": 78.930517578125,
  // The Temp_BRG_OB property is the outboard bearing temperature in degrees Fahrenheit.
  "Temp_Outboard_Bearing": 81.465283203125,
  // The Temp_Oil property is the oil temperature in degrees Fahrenheit.
  "Temp_Oil": 81.472314453125,

```

```
// The Vibe_Input_Shaft property is the vibration near the input shaft in in/sec.  
"Vibe_Input_Shaft": 0.0310514681916677,  
// The Vibe_Output_Shaft property is the vibration near the output shaft in in/sec.  
"Vibe_Output_Shaft": 0.0310834173259751,  
// The Motor_Amps property is the calculated amp draw of the input motor. This will not be  
available for all units.  
"Motor_Amps": 0,  
// The Input_Shaft_RPM property is the speed of the high speed shaft in revolutions per  
minute.  
"Input_Shaft_RPM": 0  
}
```