Power IQ

WS-API Programming Guide
Release 4.0.2

FCC Information

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a commercial installation. This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. Operation of this equipment in a residential environment may cause harmful interference.

VCCI Information (Japan)

この装置は、情報処理装置等電波障害自主規制協議会（ＶＣＣＩ）の基準
に基づくクラスＡ情報技術装置です。この装置を家庭環境で使用すると電波
妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ず
るよう要求されることがあります。

If a power cable is included with this product, it must be used exclusively for this product.

Contents

Chapter 1   Introduction to the Power IQ API

This section describes the Power IQ web services REST API.

The API is bi-directional: external applications can access and update the information in Power IQ.

In short, the document describes:

- Example use cases of what can be done with this API
- A technical description of the API
- A functional description of the API including a list of API calls, and an overview of the relevant objects and their fields

The word "PDU" in this document means "rack PDU". The concept "EDM entity" includes PDUs, Devices, Racks, Data Centers etc. EDM is an abbreviation of Enterprise Data Model.

Power IQ can be used:

- as a stand-alone application (without integration with an external application)
- integrated with an external application using the API described in this document
- a combination of both, because changes made in Power IQ will show up in the external asset management application and vice versa

The REST API also contains bulk data commands for Enterprise Power IQ. Please send an email to jamesc@raritan.com if you are interested in Enterprise Power IQ API.

In This Chapter

## Guidance for Customers Upgrading from Previous Releases

This section is intended to provide guidance to customers who use the REST API and are planning to upgrade to Power IQ release 4.0.

In Power IQ 4.0, a couple of changes were made to the system and REST API to provide granular data for the PDU inlets, better performance, allow for power control, and to make it easier to get rollup data.

- PDU Unit Level readings were replaced with more granular inlet level readings. This allows Power IQ to support PDU's with more than one inlet
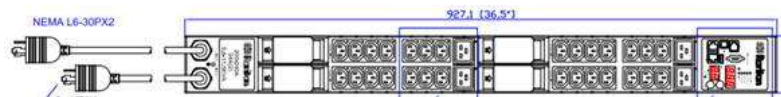
- A new inlet id field was added to lines and line readings so that users can determine which lines and line readings are associated with which inlets.
- Each of the rollup levels (monthly, daily and hourly) were given their own routes such that users no longer have to specify query parameters. The routes also provide better performance.
- New data points were exposed including voltage at the inlet pole level.
- Power Control was added to the REST API.

As a result of these changes, the following routes are no longer supported.

```
GET /api/v2/pdus/:pdu_id/readings
GET /api/v2/pdus/:pdu_id/readings_rollups

GET /api/v2/line_readings
GET /api/v2/line_readings_rollups

GET /api/v2/circuit_breaker_readings_rollups
GET /api/v2/outlet_readings_rollups
GET /api/v2/outlets/:outlet_id/readings_rollups
GET /api/v2/sensor_readings_rollups
GET /api/v2/sensors/:sensor_id/readings_rollups
```

For instructions on retrieving the equivalent data in 4.0, see:

- ***Guidance for PDU Readings in 4.0*** (on page 6)
- ***Guidance for Line Readings in 4.0*** (on page 9)
- ***Guidance for Monthly, Daily, and Hourly Rollups in 4.0*** (on page 10)

Guidance for PDU Readings in 4.0

In previous releases, customers could get PDU level data using one of the following two routes. There was no way to retrieve inlet level data.

```
GET /api/v2/pdus/:pdu_id/readings [DEPRECATED]
GET /api/v2/pdus/:pdu_id/readings_rollups [DEPRECATED]
```

In R4.0, these routes were replaced with ones that expose inlet level data.   The new routes are as follows.   Note that a new inlet id was added to lines and line readings so that users can determine which lines and line readings are associated with which inlets.

```
GET /api/v2/inlet_readings
GET /api/v2/inlet_readings_rollups/hourly
GET /api/v2/inlet_readings_rollups/daily
GET /api/v2/inlet_readings_rollups/monthly
```

To get the readings for a specific PDU, you can specify the pdu_id using the searching parameters. For example, to retrieve values for a PDU with pdu_id = 50, you would specify the following:

```
GET /api/v2/inlet_readings?pdu_id_eq=50
GET /api/v2/inlet_readings_rollups/hourly?pdu_id_eq=50
GET /api/v2/inlet_readings_rollups/daily?pdu_id_eq=50
GET /api/v2/inlet_readings_rollups/monthly?pdu_id_eq=50
```

If you query the inlet_readings for a PDU with three inlets, the response would be similar to the following one. To get the total power for the PDU, sum the three active_power values for each reading time. This value is the same as the value previously returned as a PDU unit level reading.

▶ **Example:**

```
{
  "inlet_readings": [
    {
      "id": 584940,
      "pdu_id": 50,
      "inlet_id": 178,
      "reading_time": "2013/01/31 15:01:08 -0500",
      "voltage": 204.96,
      "min_voltage": null,
      "max_voltage": null,
      "current": 15.417,
      "min_current": null,
      "max_current": null,
      "unutilized_capacity": 16.583,
      "min_unutilized_capacity": null,
      "max_unutilized_capacity": null,
      "power_factor": null,
      "min_power_factor": null,
      "max_power_factor": null,
      "active_power": 3456,
```

```
    "min_active_power": null,
    "max_active_power": null,
    "apparent_power": 3617,
    "min_apparent_power": null,
    "max_apparent_power": null,
    "volt_amp_hour": null,
    "watt_hour": 16122800
    "inlet_ordinal": 3"
},
{
    "id": 584939,
    "pdu_id": 50,
    "inlet_id": 177,
    "reading_time": "2013/01/31 15:01:08 -0500",
    "voltage": 204.96,
    "min_voltage": null,
    "max_voltage": null,
    "current": 12.274,
    "min_current": null,
    "max_current": null,
    "unutilized_capacity": 19.726,
    "min_unutilized_capacity": null,
    "max_unutilized_capacity": null,
    "power_factor": null,
    "min_power_factor": null,
    "max_power_factor": null,
    "active_power": 3268,
    "min_active_power": null,
    "max_active_power": null,
    "apparent_power": 3352,
    "min_apparent_power": null,
    "max_apparent_power": null,
    "volt_amp_hour": null,
    "watt_hour": 14569100
    "inlet_ordinal": 2"
},
{
    "id": 584938,
    "pdu_id": 50,
    "inlet_id": 176,
    "reading_time": "2013/01/31 15:01:08 -0500",
    "voltage": 206.79,
    "min_voltage": null,
    "max_voltage": null,
    "current": 4.236,
    "min_current": null,
    "max_current": null,
    "unutilized_capacity": 27.764,
    "min_unutilized_capacity": null,
    "max_unutilized_capacity": null,
    "power_factor": null,
    "min_power_factor": null,
    "max_power_factor": null,
```

7

```
                   "active_power": 425,
                   "min_active_power": null,
                   "max_active_power": null,
                   "apparent_power": 507,
                   "min_apparent_power": null,
                   "max_apparent_power": null,
                   "volt_amp_hour": null,
                   "watt_hour": 2520460
                   "inlet_ordinal": 1"
                 }
             ]
         }
```

Guidance for Line Readings in 4.0

In previous releases, customers could get line level data that was associated with the PDU as a whole using one of the following two routes. There was no way to retrieve which lines, also known as poles, were associated with which inlets in the case of a multi-inlet PDU.

```
GET /api/v2/line_readings [DEPRECATED]
GET /api/v2/line_readings_rollups [DEPRECATED]
```

In R4.0, these routes were replaced with ones that provide the association between poles (lines) and their inlets. The new routes are as follows. These provide the same type of information as the previous line_readings calls, with the addition of a new inlet_id field so that users can determine which poles (lines) and pole (line) readings are associated with which inlets. The inlet_ordinal shows the inlet number relative to that PDU, e.g. whether it's the first, second, inlet and so on, on that PDU. Most PDU's will only have one inlet so the inlet_ordinal will be 1 on those PDU's.

```
GET /api/v2/inlet_pole_readings
GET /api/v2/inlet_pole_readings_rollups/hourly
GET /api/v2/inlet_pole_readings_rollups/daily
GET /api/v2/inlet_pole_readings_rollups/monthly
```

▶ **Example of the returned data for inlet pole readings:**

```
"inlet_pole_readings": [
        {
          "id": 754,
          "reading_time": "2002/03/27 08:30:52 +0000",
          "current": 0,
          "unutilized_capacity": 12,
          "pdu_id": 24,
          "max_current": null,
          "min_current": null,
          "inlet_pole_id": 26,
          "voltage": 124,
          "min_voltage": null,
          "max_voltage": null,
          "min_unutilized_capacity": null,
          "max_unutilized_capacity": null,
          "inlet_id": 16,
          "inlet_ordinal": 1,
          "inlet_pole_ordinal": 1
        }
    ]
```

To get the pole readings for a specific PDU, you can specify the pdu_id using the searching parameters.   For example, to retrieve values for a PDU with pdu_id =50, you would specify the following:

```
GET /api/v2/inlet_pole_readings?pdu_id_eq=50
GET /api/v2/inlet_pole_readings_rollups/hourly?pdu_id_eq=50
GET /api/v2/inlet_pole_readings_rollups/daily?pdu_id_eq=50
GET /api/v2/inlet_pole_readings_rollups/monthly?pdu_id_eq=50
```

If you want to find out all of the poles on an inlet and then retrieve the readings, you would first call the inlets route and specify the inlet_id for the inlet in question. For instance, if you wanted to find all of the poles associated with inlet_id = 16, you would call:

```
GET /api/v2/inlet_poles?inlet_id_eq=16
```

Then, for each inlet pole "id" that's returned, you would make a call to inlet_pole_readings. There are two ways to do this.

Examples for a inlet_pole_id = 26:

```
GET /api/v2/inlet_pole_readings?inlet_pole_id_eq=26
GET /api/v2/inlet_poles/26/readings
```

Guidance for Monthly, Daily, and Hourly Rollups in 4.0

This section explains how to get rollup data for customers who currently use one of the following routes:

```
/api/v2/circuit_breaker_readings_rollups [DEPRECATED]
/api/v2/line_readings_rollups [DEPRECATED]
/api/v2/outlet_readings_rollups [DEPRECATED]
/api/v2/outlets/:outlet_id/readings_rollups [DEPRECATED]
/api/v2/sensor_readings_rollups [DEPRECATED]
/api/v2/sensors/:sensor_id/readings_rollups [DEPRECATED]
```

Prior to R4.0, users could retrieve rollup data for outlet using one of the following commands.   These would return hourly, daily, or monthly rollup data respectively.

```
/api/v2/outlet_readings_rollups? rollup_interval_eq=1 [DEPRECATED]
/api/v2/outlet_readings_rollups? rollup_interval_eq=2 [DEPRECATED]
/api/v2/outlet_readings_rollups? rollup_interval_eq=3 [DEPRECATED]
```

In R4.0, the new way to get the rollup data is with these commands.

```
GET   /api/v2/outlet_readings_rollups/hourly
GET   /api/v2/outlet_readings_rollups/daily
GET   /api/v2/outlet_readings_rollups/monthly
```

To retrieve the data by outlet_id, the following options are available.

```
/api/v2/outlets/:outlet_id/readings_rollups/hourly
/api/v2/outlets/:outlet_id/readings_rollups/daily
/api/v2/outlets/:outlet_id/readings_rollups/monthly
/api/v2/outlet_readings_rollups/hourly?outlet_id_eq=:outlet_id
/api/v2/outlet_readings_rollups/daily?outlet_id_eq=:outlet_id
/api/v2/outlet_readings_rollups/monthly?outlet_id_eq=:outlet_id
```

Similar changes were made for rollups for circuit breakers, lines and sensors. See *Addendum B: Route Reference* (on page 119) for a list of the available routes.

Chapter 2   REST API

The Power IQ REST API consists of logical resources, and actions that can be taken on them. A resource is a representation of the models that are used to describe the Power IQ platform through the API. A comprehensive list of available resources is provided in **_Addendum A: Resource Reference_** (on page 48).   For example, there is a resource called pdu, and with it you can list all pdus, look at a single pdu, or make changes to a pdu.

In This Chapter

Security

To access the API calls, your application needs to authenticate itself. The REST API uses Basic Access Authentication (see http://en.wikipedia.org/wiki/Basic_access_authentication). Every HTTP request should include an extra request header with the Base64 encoded username and password. The API always uses HTTPS, so all information is transmitted encrypted, including username and password.

For example, if your username is "admin" and your password is "raritan", include in every HTTP request the following ("YWRtaW46cmFyaXRhbg==" is Base64 of "admin:raritan").

```
GET /api/v2/pdus
HTTP/1.1
Host: localhost
Authorization: Basic YWRtaW46cmFyaXRhbg==
```

Most programming languages and libraries provide built-in support for Basic Access Authentication.

Requests

All request bodies must be in JSON format, when a request body is required. Your client must set the Content-Type and Accept header of the HTTP request to application/json regardless of whether the request has body content or not, otherwise the request will not be completed successfully.

See *Addendum A: Resource Reference* (on page 48) for details on the JSON objects returned for the various available resources.

Always set the Content-Type header for requests to application/json.

Always set the Accept header for requests to application/json.

Some requests provide an option for synchronous and asynchronous modes. In these situations, pay particular attention when choosing synchronous mode over asynchronous. It is very possible that the request may time out.

A request will timeout server-side after 600s in duration. Keep in mind that individual HTTP client libraries may have lower request timeout settings.

Responses

All responses from the Power IQ API will be in JSON format. Appropriate HTTP response codes are also used to indicate the results of an action on a resource.

See *Addendum E: Response Codes* (on page 128) below for details on the value and meanings of various response codes.

Everything but the smallest responses are GZip encoded. Make sure you client has support for GZip.

Testing

In order to test and explore the Power IQ REST API, it can be very helpful to use FireFox with the RESTClient plugin. To set this up do the following:

1. Install the latest version of FireFox (http://www.getfirefox.net)

EXE Raritan.

2. Install the REST Client FireFox plugin
   (https://addons.mozilla.org/en-us/firefox/addon/)

3. Accept certificates

4. Open up the location of Power IQ in FireFox (not in the REST Client)

5. Permanently accept all certificates

6. Open up the REST Client plug-in in FireFox

7. Start FireFox

8. Select Tools > REST Client

9. Setup basic authentication

10. From the REST Client interface, select the Login button at the top

11. In the login dialog select Basic, and enter in the appropriate Login
    and Password values

12. Select OK to dismiss the dialog box

13. Setup headers

14. From the REST Client interface, select the Add Request Header
    button at the top

15. Set the Name field to Content-Type

16. Set the Value field to application/json

17. Select OK to apply

18. From the REST Client interface, select the Add Request Header
    button at the top

19. Set the Name field to Accept

20. Set the Value field to application/json

21. Select OK to apply

22. Make your first request

23. From the REST Client interface, input the URL for your Power IQ
    server to retrieve a listing of all PDUs: http://<server>/api/v2/pdus

24. In the Method drop-down, select GET

25. Select the Send button to make the request

26. The panels at the bottom provide different views into the request and
    response

Functional Areas

The Power IQ Restful API can be grouped into the following general areas of functionality:

- *Job Management* (on page 16)
- *Event Management* (on page 18)
- *Outlet Management* (on page 20)
- *Data Center Hierarchy* (on page 26)
- *Asset Strip Management* (on page 32)
- *PDU Management* (on page 34)
- *Power and sensor readings* (on page 44)
- *Power Control* (on page 45)
- *Misc* (on page 47)

API

The resources and actions listed below constitute the Power IQ REST API. Please observe the following general guidelines:

GET methods never require any content in the body of the request

Assume the request and response body for a given resource will be in the JSON format as defined in *Addendum A: Resource Reference* (on page 48) for that resource, unless otherwise noted

PUT, POST, and DELETE methods almost always require a JSON body to be present in the request

If the async parameter is set to true on a request that supports it, the response body will be a job resource

Errors

Power IQ uses standard HTTP response codes to indicate the success or failure of an API request. For the most part, codes in the 2xx range indicate success, codes in the 4xx range indicate an error in the provided information, and codes in the 5xx range indicate an error within Power IQ. For more details on these codes, see *Addendum E: Response Codes* (on page 128).

In addition to HTTP response codes, Power IQ will usually include JSON in the response body of the request to help detail the problem that occurred.

Example:

▶ **REQUEST**

```
Content-Type: application/json
Accept: application/json; charset=utf-8
POST /api/v2/devices

{ "device" : { "parent" : { "data_center" : 1 } } }
```

▶ **RESPONSE**

```
Status Code: 400 Bad Request
Content-Type: application/json; charset=utf-8

{
  "error":"ActiveRecord::RecordInvalid",
  "messages":["Name can't be blank"]
}
```

The response code 400 informs the client that there was a problem with the request. The JSON body response details the error that occurred. In this case, the request was missing the name attribute for the device resource.

Job Management

Job resources are used to handle long running requests that either must be, or can optionally be, executed asynchronously. An action on any resource may return a job resource if it supports the async parameter.

▶ **Route Reference:**

```
GET   /api/v2/jobs/:id
GET   /api/v2/jobs/:job_id/messages
GET   /api/v2/job_messages/:id
GET   /api/v2/job_messages
```

GET /api/v2/jobs/:id

Returns a single job, specified by the id parameter below.

Parameters

id	The numerical ID of the desired job.
required	Example Values: 123

GET /api/v2/jobs/:job_id/messages

Returns all messages associated with the given job, specified by the job_id parameter below. Job messages contained detailed information about the invidual steps during the execution of a job, including detailed error messages if an error exists.

Parameters

job_id	The numerical ID of the desired job that messages should be retrieved for.
required	Example Values: 123

GET /api/v2/job_messages/:id

Returns a single job_message, specified by the id parameter below.

Parameters

id	The numerical ID of the desired job_message.
required	Example Values: 123

GET /api/v2/job_messages

Returns a list of all job_messages.

Parameters

* searchable	This request supports searching. See **Addendum C: Searching Reference** (on page 125) and **Addendum A: Resource Reference** (on page 48) to see which fields this resource can be searched on.

Event Management

Events are typically traps collected by Power IQ from PDUs and put into a normalized form, but some can also be created internally by Power IQ.

▶ **Route Reference**

```
GET    /api/v2/events/:id
GET    /api/v2/events
PUT    /api/v2/events/clear_batch
PUT    /api/v2/events/:id/clear
```

GET /api/v2/events/:id

Returns a single event, specified by the id parameter below.

Parameters

id	The numerical ID of the desired event.
required	Example Values: 123

GET /api/v2/events

Returns a listing of all events

Parameters

*	This request supports searching. See **Addendum C: Searching Reference** (on page 125) and **Addendum A: Resource Reference** (on page 48) to see which fields this resource can be searched on.
searchable	

PUT /api/v2/events/clear_batch

Clears a list of events in the system by event id, acknowledging receipt of that event and allowing it to be pruned by the system.

▶ **Request Body**

```
{ "events" : [ 1, 2, 26 ] }
```

▶ **Response Body**

```
{"events":[
    {
        "id":1,
        "event_config_id":72,
        "source":2,
        "created_at":"2011/10/07 14:49:42 +0000",
        "pdu_id":20,
```

```
        "pdu_outlet_id":null,
        "pdu_circuitbreaker_id":null,
        "sensor_id":null,
        "trap_oid":null,
        "cleared_by":2,
        "cleared_at":"2011/10/25 17:47:18 +0000",
        "clearing_event_id":null,
        "clearing_user_id":1,
        "notification_status":6,
        "asset_strip_id":null,
        "rack_unit_id":null,
        "params":[]
    },
    {
        "id":2,
        "event_config_id":72,
        "source":2,
        "created_at":"2011/10/07 14:49:45 +0000",
        "pdu_id":23,
        "pdu_outlet_id":null,
        "pdu_circuitbreaker_id":null,
        "sensor_id":null,
        "trap_oid":null,
        "cleared_by":2,
        "cleared_at":"2011/10/25 17:47:18 +0000",
        "clearing_event_id":null,
        "clearing_user_id":1,
        "notification_status":6,
        "asset_strip_id":null,
        "rack_unit_id":null,
        "params":[]
    },
    {
        "id":26,
        "event config id":62,
        "source":2,
        "created_at":"2011/10/13 15:05:55 +0000",
        "pdu_id":22,
        "pdu_outlet_id":null,
        "pdu_circuitbreaker_id":null,
        "sensor_id":null,
        "trap_oid":null,
        "cleared_by":2,
        "cleared_at":"2011/10/25 17:47:18 +0000",
        "clearing_event_id":null,
        "clearing_user_id":1,
        "notification_status":6,
        "asset_strip_id":null,
        "rack_unit_id":null,
        "params":[
            {
                "key":"timestamp",
```

```
            "value":"1318513900627"
          }
        ]
      }
]}
```

PUT /api/v2/events/:id/clear

Clears a single event in the system, acknowledging receipt of that event and allowing it to be pruned by the system.

Parameters

id	The numerical ID of the event to be cleared.
required	Example Values: 123

Outlet Management

Outlets are components of a PDU and can be associated to a device resource.

Route Reference

GET	/api/v2/outlet_readings
GET	**/api/v2/outlet_readings_rollups**
GET	/api/v2/outlet_readings_rollups/hourly
GET	/api/v2/outlet_readings_rollups/daily
GET	/api/v2/outlet_readings_rollups/monthly
GET	/api/v2/outlets
GET	/api/v2/outlets/:id
PUT	/api/v2/outlets/:id
GET	/api/v2/outlets/:outlet_id/readings
GET	**/api/v2/outlets/:outlet_id/readings_rollups [DEPRECATED]**
GET	/api/v2/outlets/:outlet_id/readings_rollups/hourly
GET	/api/v2/outlets/:outlet_id/readings_rollups/daily
GET	/api/v2/outlets/:outlet_id/readings_rollups/monthly
GET	/api/v2/outlets/:outlet_id/events

POST	/api/v2/outlets/power_control
PUT	/api/v2/outlets/rename_batch

GET /api/v2/outlets/:id

Returns a single outlet, specified by the id parameter below.

Parameters

id	The numerical ID of the desired outlet.
required	Example Values: 123

GET /api/v2/outlets

Returns a listing of all outlets

Parameters

*	This request supports searching. See ***Addendum C: Searching Reference*** (on page 125) and ***Addendum A: Resource Reference*** (on page 48) to see which fields this resource can be searched on.
searchable	

PUT /api/v2/outlets/:id

Update the outlet, specified by the id parameter below. This method can be used to rename a single outlet and re-assign an outlet to a Device. To rename multiple outlets use rename_batch.

Parameters

id	The numerical ID of the desired outlet.
required	Example Values: 123

▶ **Request Body**

```
{"outlet":{
    "outlet_name":"NewTestOutletLabel",
    "device_id":null,
}}
```

▶ **Response Body**

```
{"outlet":{
    "id":1,
    "outlet_id":1,
    "outlet_name":"NewTestOutletLabel",
    "device_id":null,
    "state":"ON",
    "pdu_id":15
}}
```

≡≡Raritan.

PUT /api/v2/outlets/rename_batch

Rename multiple outlets.

Parameters

async asynchronous	This request supports synchronous and asynchronous modes of operation.
	Example Value(s): true
	Valid Values: (false|true)

► **EXAMPLE 1:**

► **Request**

```
Content-Type: application/json
Accept: application/json; charset=utf-8
PUT /api/v2/outlets/rename_batch?async=false

{"outlets":[
   {"id": 1, "outlet_name": "label1"},
   {"id": 2, "outlet_name": "label2"},
   {"id": 3, "outlet_name": "label3"}
]}
```

► **Response**

```
Status: 200
Content-Type: application/json; charset=utf-8

{"outlets":[
    {
        "id":1,
        "outlet_id":1,
        "outlet_name":"label1",
        "device_id":null,
        "state":"ON",
        "pdu_id":15,
        "reading":{}
    },
    {
        "id":2,
        "outlet_id":2,
        "outlet_name":"label2",
        "device_id":null,
        "state":"ON",
        "pdu_id":15,
        "reading":{}
    },
    {
```

```
        "id":3,
        "outlet_id":3,
        "outlet_name":"label3",
        "device_id":null,
        "state":"ON",
        "pdu_id":15,
        "reading":{}
    }
]}
```

▶ **EXAMPLE 2:**

▶ **Request**

```
Content-Type: application/json
Accept: application/json; charset=utf-8
PUT /api/v2/outlets/rename_batch?async=true

{"outlets":[
    {"id": 1, "outlet_name": "label1"},
    {"id": 2, "outlet_name": "label2"},
    {"id": 3, "outlet_name": "label3"}
]}
```

▶ **Response**

```
Status: 200
Content-Type: application/json; charset=utf-8

{"job":{
    "id":1,
    "user_id":1,
    "status":"COMPLETED",
    "description":null,
    "start_time":"2011/10/07 14:54:32 +0000",
    "end_time":"2011/10/07 14:54:33 +0000",
    "has_errors":false,
    "percent_complete":1.0,
    "completed":true,
    "last_message":"outlet ID=3 renamed to label3",
    "error_count":0
}}
```

GET /api/v2/outlets/:outlet_id/events

Retrieve any events associated with the given outlet.

Parameters

outlet_id	The numerical ID of the desired outlet.
required	Example Values: 123

GET /api/v2/outlets/:outlet_id/readings

Retrieve latest readings associated with the given outlet.

Parameters

outlet_id	The numerical ID of the desired outlet.
required	Example Values: 123

GET /api/v2/outlets/:outlet_id/readings_rollups/:rollup_interval

Retrieve different intervals of historical rollup readings associated with the given outlet.

Parameters

outlet_id	The numerical ID of the desired outlet.
required	Example Values: 123
rollup_interval	The time interval of historical rollup readings to retrieve.
required	Example Value: hourly
	Valid Values: hourly | daily | monthly

Data Center Hierarchy

The data center hierarchy in Power IQ is a tree data structure that represents the resources associated with a typical data center. This includes resources such as racks, rows, rooms, and IT devices. This portion of the API provides access to those resources, as well as ways to navigate the hierarchy.

Although PDUs are a part of the data center hierarchy, the API for PDUs is significantly different than other data center resources, and is documented in the PDU Management section.

▶ **Route Reference**

```
GET    /api/v2/aisles
POST   /api/v2/aisles
GET    /api/v2/aisles/:id
PUT    /api/v2/aisles/:id
DELETE /api/v2/aisles/:id
GET    /api/v2/aisles/:id/children
GET    /api/v2/aisles/:id/descendants
GET    /api/v2/aisles/:id/siblings
PUT    /api/v2/aisles/:id/move_to

GET    /api/v2/data_centers
POST   /api/v2/data_centers
GET    /api/v2/data_centers/:id
PUT    /api/v2/data_centers/:id
DELETE /api/v2/data_centers/:id
GET    /api/v2/data_centers/:id/children
GET    /api/v2/data_centers/:id/descendants
GET    /api/v2/data_centers/:id/siblings
PUT    /api/v2/data_centers/:id/move_to

GET    /api/v2/devices
POST   /api/v2/devices
GET    /api/v2/devices/:id
PUT    /api/v2/devices/:id
DELETE /api/v2/devices/:id
GET    /api/v2/devices/:id/siblings
PUT    /api/v2/devices/:id/move_to
GET    /api/v2/devices/:device_id/outlets

GET    /api/v2/floors
POST   /api/v2/floors
GET    /api/v2/floors/:id
PUT    /api/v2/floors/:id
DELETE /api/v2/floors/:id
GET    /api/v2/floors/:id/children
GET    /api/v2/floors/:id/descendants
GET    /api/v2/floors/:id/siblings
```

```
PUT    /api/v2/floors/:id/move_to

GET    /api/v2/racks
POST   /api/v2/racks
GET    /api/v2/racks/:id
PUT    /api/v2/racks/:id
DELETE /api/v2/racks/:id
GET    /api/v2/racks/:id/children
GET    /api/v2/racks/:id/descendants
GET    /api/v2/racks/:id/siblings
PUT    /api/v2/racks/:id/move_to

GET    /api/v2/rooms
POST   /api/v2/rooms
GET    /api/v2/rooms/:id
PUT    /api/v2/rooms/:id
DELETE /api/v2/rooms/:id
GET    /api/v2/rooms/:id/children
GET    /api/v2/rooms/:id/descendants
GET    /api/v2/rooms/:id/siblings
PUT    /api/v2/rooms/:id/move_to

GET    /api/v2/rows
POST   /api/v2/rows
GET    /api/v2/rows/:id
PUT    /api/v2/rows/:id
DELETE /api/v2/rows/:id
GET    /api/v2/rows/:id/children
GET    /api/v2/rows/:id/descendants
GET    /api/v2/rows/:id/siblings
PUT    /api/v2/rows/:id/move_to
```

GET /api/v2/:resource/:id

Returns a single data center resource, specified by the id and resource parameters below.

Parameters

id	The numerical ID of the desired resource.
required	Example Values: 123
resource	The name of the data center resource.
required	Example Values: data_centers
	Valid Values: (aisles|data_centers|devices|floors|racks|rooms|rows)

PUT /api/v2/:resource/:id

27

Update a data center resource, specified by the id and resource parameters below.

Parameters

id	The numerical ID of the desired resource.
required	Example Values: 123
resource	The name of the data center resource.
required	Example Values: data_centers
	Valid Values: (aisles|data_centers|devices|floors|racks|rooms|rows)

DELETE /api/v2/:resource/:id

Delete a data center resource, specified by the id and resource parameters below.

Deleting a data center resource delete's that resource and all it's descendants.

Parameters

id	The numerical ID of the desired resource.
required	Example Values: 123
resource	The name of the data center resource.
required	Example Values: data_centers
	Valid Values: (aisles|data_centers|devices|floors|racks|rooms|rows)

POST /api/v2/:resource

Create a new data center resource. The parent of the resource must be specified, such that this resource is added to the correct place within the data center hierarchy.

Parameters

resource	The name of the data center resource.
required	Example Values: data_center
	Valid Values: (aisles|data_centers|devices|floors|racks|rooms|rows)

▶ **Request**

```
Content-Type: application/json
Accept: application/json; charset=utf-8
POST /api/v2/rooms

{ "room" : {
    "name" : "Big Room",
    "external_key" : "AAFF",
    "parent" : {
        "data_center" : {
            "id" :2
        }
    }
} }
```

▶ **Response**

```
Status: 200
Content-Type: application/json; charset=utf-8

{"room":{
    "id":2,
    "name":"Big Room",
    "external_key":"AAFF",
    "capacity":null
}}
```

GET /api/v2/:resource

Returns all of the specified data center resources, specified by the resource parameter.

Parameters

resource	The name of the data center resource.
required	Example Values: data_centers
searchable	Valid Values: (aisles|data_centers|devices|floors|racks|rooms|rows)

GET /api/v2/:resource/:id/children

Return all the children of the data center resource, specified by the resource and id parameters.

Parameters

29

id	The numerical ID of the desired resource.
required	Example Values: 123

resource	The name of the data center resource.
required	Example Values: data_centers
	Valid Values: (aisles|data_centers|floors|racks|rooms|rows)

Devices have no children, so device is not a valid option for the resource parameter.

GET /api/v2/:resource/:id/descendants

Return all the descendants of the data center resource, specified by the resource and id parameters.

Parameters

id	The numerical ID of the desired resource.
required	Example Values: 123

resource	The name of the data center resource.
required	Example Values: data_centers
	Valid Values: (aisles|data_centers|floors|racks|rooms|rows)

Devices have no descendants, so device is not a valid option for the resource parameter.

GET /api/v2/:resource/:id/siblings

Return all the siblings of the data center resource, specified by the resource and id parameters.

Parameters

id	The numerical ID of the desired resource.
required	Example Values: 123

resource	The name of the data center resource.
required	Example Values: data_centers
	Valid Values: (aisles|data_centers|devices|floors|racks|rooms|rows)

PUT /api/v2/:resource/:id/move_to

Move the data center resource to another location, specified by the resource and id parameters. The body of the request must include the location to which the resource should be moved.

Parameters

id	The numerical ID of the desired resource.
required	Example Values: 123
resource	The name of the data center resource.
required	Example Values: data_centers
	Valid Values: (aisles|data_centers|devices|floors|racks|rooms|rows)

▶ **Request**

```
Content-Type: application/json
Accept: application/json; charset=utf-8
PUT /api/v2/rooms/2/move_to

{ "data_center" : { "id" : 2 } }
```

GET /api/v2/device/:device_id/outlets

Retrieve the outlets associated with the device.

Parameters

device_id	The numerical ID of the desired device.
required	Example Values: 123

Asset Strip Management

Asset strips and rack units represent EMX asset strips and the individual LEDS on the asset strip, respectively.

▶ **Route Reference**

```
GET    /api/v2/asset_strips
GET    /api/v2/asset_strips/:id
PUT    /api/v2/asset_strips/:id
GET    /api/v2/asset_strips/:asset_strip_id/rack_units

GET    /api/v2/rack_units
GET    /api/v2/rack_units/:id
PUT    /api/v2/rack_units/:id
GET    /api/v2/rack_units/:id/blade_slots

GET /api/v2/blade_slots
GET /api/v2/blade_slots/:id
```

GET /api/v2/asset_strips/:id

Returns a single asset_strip resource, specified by the id parameter below. An asset_strip resource has a collection of rack_units that represent the LED's attached to it.

Parameters

id	The numerical ID of the desired asset_strip.
required	Example Values: 123

PUT /api/v2/asset_strips/:id

Update a asset_strip resource, specified by the id parameter below.

Parameters

id	The numerical ID of the desired asset_strip.
required	Example Values: 123

GET /api/v2/asset_strips/:asset_strip_id/rack_units

Return a listing of all LEDs (rack_units) associated with the asset_strip_id specified in the parameter listing below.

Parameters

asset_strip_id	The numerical ID of the desired asset_strip.

required	Example Values: 123

GET /api/v2/asset_strips

Returns all asset_strip resources.

Parameters

* searchable	This request supports searching. See Addendum C: Searching Reference and Addendum A: Resource Reference to see which fields this resource can be searched on.

GET /api/v2/rack_units/:id

Returns a single rack_unit resource, specified by the id parameter below. A rack_unit represents an individual LED on an asset_strip resource.

Parameters

id	The numerical ID of the desired rack_unit.
required	Example Values: 123

PUT /api/v2/rack_units/:id

Updates a single rack_unit resource, specified by the id parameter below.

Parameters

id	The numerical ID of the desired rack_unit.
required	Example Values: 123

GET /api/v2/rack_units

Returns all rack_units resources.

Parameters

* searchable	This request supports searching. See **Addendum C: Searching Reference** (on page 125) and **Addendum A: Resource Reference** (on page 48) to see which fields this resource can be searched on.

GET /api/v2/rack_units/:id/blade_slots

Returns all blade slots attached to the rack unit defined by that id.

Parameters

id	The numerical ID of the desired rack_unit.
required	Example Values: 123

GET /api/v2/blade_slots

Returns ALL blade slots resources.

Parameters

* searchable	This request supports searching. See Addendum C: Searching Reference and Addendum A: Resource Reference to see which fields this resource can be searched on.

GET /api/v2/blade_slots/:id

Returns a single blade_slot resource, specified by the id parameter

id	The numerical ID of the desired blade_slot.
required	Example Values: 123

PDU Management

A PDU resource is central to most aspects of the Power IQ API. Although PDUs participate in the data center hierarchy, they have many unique operations that set them apart from other data center resources.

▶ **Route Reference**

POST /api/v2/pdus/create_batch

DELETE /api/v2/pdus/destroy_batch

GET /api/v2/pdus

POST /api/v2/pdus

GET /api/v2/pdus/:id

PUT /api/v2/pdus/:id

DELETE /api/v2/pdus/:id

GET /api/v2/pdus/:id/inlets

PUT /api/v2/pdus/:id/move_to

PUT /api/v2/pdus/:id/rescan

GET   /api/v2/pdus/:pdu_id/asset_strips

GET   /api/v2/pdus/:pdu_id/events

GET   /api/v2/pdus/:pdu_id/outlets

GET   /api/v2/pdus/:pdu_id/sensors

GET /api/v2/pdus

Returns all pdu resources.

Parameters

* searchable	This request supports searching. See **Addendum C: Searching Reference** (on page 125) and **Addendum A: Resource Reference** (on page 48) to see which fields this resource can be searched on.

POST /api/v2/pdus

Create a pdu resource.

A pdu creation request should contain the IP Address, proxy index (if needed), and the access information.   For Raritan PDUs this would include the userid and password required for upgrades.   SNMPv1 or SNMPv2 will require an appropriate community string while SNMPv3 is much more complex.

▶ **Valid fields**

ip_address -- The IP Address of the PDU

ipmi_username -- The username of the user used to log into the Web UI of the PDU if it is a Raritan PDU

ipmi_password -- The password of the user used to log into the Web UI of the PDU if it is a Raritan PDU

proxy_index -- The identifying index of a PDU that is under the control of a central system.   The central system would have one IP address and each PDU would have a proxy indice.

snmp3_enabled -- "true" or "false", indicates if SNMPv3 is enabled or not.

snmp_community_string -- The SNMP v1 and v2 write community string used by PowerIQ to get and set data on the PDU.

snmp3_user -- The SNMPv3 user

snmp3_auth_level -- The SNMP v3 authorization level: "noAuthNoPriv", "authNoPriv" or "authPriv"

snmp3_auth_passkey -- The SNMPv3 authorization passkey

snmp3_auth_protocol -- The SNMPv3 authorization protocol: "MD5" or "SHA"

snmp3_priv_passkey -- The SNMPv3 privacy passkey

snmp3_priv_protocol -- The SNMPv3 privacy protocol: "DES" or "AES"

▶ **Request Body**

```
For a Raritan PDU:
POST /api/v2/pdus
{ "pdu" : { "ip_address" : "192.168.1.1",
"ipmi_username" : "admin-user", "ipmi_password" :
"password", "snmp3_enabled" : "false",
"snmp_community_string" : "private"} }

For a non-Raritan PDU:
POST /api/v2/pdus
{ "pdu" : { "ip_address" : "192.168.1.1",
"snmp3_enabled" : "false", "snmp_community_string" :
"private"} }

For a PDU with a proxy controller:
POST /api/v2/pdus
{ "pdu" : { "ip_address" : "192.168.1.1",
"proxy_index" : "1", "snmp3_enabled" : "false",
"snmp_community_string" : "private"} }
```

▶ **Response Body**

```
Successful response:

{
 "pdu" :
  {
   "id":1,
   "snmp3_enabled":false,
   "snmp3_user":null,
   "snmp3_auth_level":null,
   "caption":"PX-3370",
   "description":"Raritan Dominion PX - Firmware Version
010500-10315",
   "contact":"Test Lab",
   "proxy_index":null,
   "requires_manual_voltage":null,
   "configured_inlet_voltage":null,
   "configured_outlet_voltage":null,
   "supports_single_sign_on":true,
   "supports_firmware_upgrades":true,
   "supports_bulk_configuration":true,
   "supports_outlet_power_control":true,
   "supports_outlet_renaming":true,
   "name":"PX-3370",
   "model":"PX-3370",
   "location":"Test Lab",
   "serial_number":"PTI0390259",
```

```
   "manufacturer":"Raritan",
   "firmware_version":"1.5.0-10315",

"poller_plugin":"com.raritan.polaris.plugins.pdu.ra
ritan.PduPoller",
   "rated_volts":"400V",
   "rated_amps":"32.00A",
   "rated_va":"22.0kVA",
   "ip_address":"192.168.1.1",
   "inline_meter":true,
   "supports_readingsonly_poll":false,
   "supports_data_logging":true,
   "supports_sensor_renaming":true,
   "default_connected_led_color":null,
   "default_disconnected_led_color":null,
   "dynamic_plugin_name":null,
   "phase":"UNKNOWN",
   "user_defined_phase":false,
   "custom_field_1":null,
   "custom_field_2":null,
   "external_key":"192.168.1.1",
"reading": {
  "inlet_readings": [
    {
      "inlet_id": 7,
      "volt_amp_hour": null,
      "min_unutilized_capacity": null,
      "max_power_factor": null,
      "min_apparent_power": null,
      "voltage": 120,
      "active_power": 98,
      "min_voltage": null,
      "pdu_id": 16,
      "max_current": null,
      "current": 1.193,
      "min_current": null,
      "max_apparent_power": null,
      "unutilized_capacity": 10.807,
      "watt_hour": null,
      "power_factor": 0.705036,
      "id": 14549,
      "max_unutilized_capacity": null,
      "max_voltage": null,
      "min_power_factor": null,
      "max_active_power": null,
      "apparent_power": 139,
      "reading_time": "2012/09/11 14:12:58 -0400",
      "min_active_power": null
    }
  ],
  "inlet_pole_readings": [
    {
```

```
      "id": 21833,
      "current": 1.193,
      "pdu_id": 16,
      "voltage": 120,
      "reading_time": "2012/09/11 14:12:58 -0400",
      "min_current": null,
      "unutilized_capacity": 10.807,
      "inlet_pole_id": 7,
      "max_unutilized_capacity": null,
      "min_voltage": null,
      "max_current": null,
      "max_voltage": null,
      "min_unutilized_capacity": null
   }
 ],
 "circuit_breaker_readings": [
 ]
}


   },
   "health":
    {
    "overall":"Good",
    "connectivity":"OK",
    "connectivity_explanation":"Most recent poll of
the target PDU was successful.",
    "events":"Good",
    "active_events_count":0
    }
  }
}


Error response:

{
 "error":"Job::JobError",
 "messages":
  [
   "Job(ID:91) with 2 error(s) and status COMPLETED
completed",
   "Queuing PDU 192.168.1.1 for discovery",
   "Beginning discovery for PDU 192.168.1.1 ",
   "Discovering PDU 192.168.1.1 and could not retrieve
SystemObjectID",
   "PDU 192.168.1.1 could not be discovered.  Failure
code: NoPlugin",
   "PDU 192.168.1.1 could not be discovered.  There will
be no poll."
  ],
  "job":
   {
```

```
    "description":null,
    "end_time":"2011/10/24 17:29:16 -0400",
    "has_errors":true,
    "id":91,
    "start_time":"2011/10/24 17:28:55 -0400",
    "status":"COMPLETED",
    "user_id":1
  },
  "trace":"Data here will be valuable only to Raritan
Tech support."
}
```

GET /api/v2/pdus/:id

Retrieve a single pdu resource, specified by the id parameter below.

Parameters

id	The numerical ID of the desired pdu.
required	Example Values: 123

PUT /api/v2/pdus/:id

Update a single pdu resource, specified by the id parameter below.

Parameters

id	The numerical ID of the desired pdu.
required	Example Values: 123

DELETE /api/v2/pdus/:id

Delete a single pdu resource, specified by the id parameter below.

Once a PDU is removed, it's readings are permanently lost and cannot be recovered.

Parameters

id	The numerical ID of the desired pdu.
required	Example Values: 123

GET /api/v2/pdus/:id/inlets

Retrieve all inlets associated with the PDU resource specified by the id parameter below.

Parameters

id	The numerical ID of the desired pdu.
required	Example Values: 123

GET /api/v2/pdus/:id/rescan

Rescan a single pdu resource, specified by the id parameter below. PDUs are remotely managed devices polled based on the settings in Power IQ. By requesting a rescan of a PDU, that device will be scheduled for immediate polling, providing the latest data for that PDU.

Parameters

id	The numerical ID of the desired pdu.
required	Example Values: 123

PUT /api/v2/pdus/:id/move_to

Move the pdu resource to another location, specified by the id parameter. The body of the request must include the location to which the pdu should be moved too.

Parameters

id	The numerical ID of the desired pdu that should be moved.
required	Example Values: 123

▶ **Request Body**

{"rack":{"id":"1"}}

POST /api/v2/pdus/create_batch

Creates multiple pdu resources.   This is always done in an asynchronous manner, so a search must be done to find the PDU once the job has completed.   One may use the job api to gather information about the job id returned.   In the sample below we combine the examples from the single create method and have one request for creating a batch of PDUs.

▶ **Request Body**

```
POST /api/v2/pdus/create_batch

{ "pdus" : [{ "ip_address" : "192.168.1.1",
"ipmi_username" : "admin-user", "ipmi_password" :
"password", "snmp3_enabled" : "false",
"snmp_community_string" : "private"},{ "ip_address" :
"192.168.1.2", "snmp3_enabled" : "false",
"snmp_community_string" : "private"}, { "ip_address" :
"192.168.1.3", "proxy_index" : "1", "snmp3_enabled" :
"false", "snmp_community_string" : "private"}] }
```

► **Response Body**

```
{"job":{"id":94,"user_id":1,"status":"ACTIVE","desc
ription":null,"start_time":"2011/10/25 11:48:08
-0400","end_time":null,"has_errors":false,"percent_
complete":0.0,"completed":false,"last_message":"Que
uing PDU 192.168.1.1 for discovery","error_count":0}}
```

DELETE /api/v2/pdus/destroy_batch

Destroy multiple pdu resources. This is done based on ID.

► **Request Body**

```
{"pdus":["1","2","3"]}
```

► **Response Body**

```
{}
```

GET /api/v2/pdus/:id/sensors

Retrieve all sensors associated with the pdu resource specified by the id
parameter below.

Parameters

id	The numerical ID of the desired pdu.
required	Example Values: 123

GET /api/v2/pdus/:id/outlets

Retrieve all outlets associated with the pdu resource specified by the id
parameter below.

Parameters

id	The numerical ID of the desired pdu.
required	Example Values: 123

GET /api/v2/pdus/:id/asset_strips

Retrieve all asset_strips associated with the pdu resource specified by the id parameter below.

Parameters

id	The numerical ID of the desired pdu.
required	Example Values: 123

GET /api/v2/pdus/:id/events

Retrieve all events associated with the pdu resource specified by the id parameter below.

Parameters

id	The numerical ID of the desired pdu.
required	Example Values: 123

Power and sensor readings

Readings are powerful metrics collected by Power IQ. In the Power IQ web UI, readings are the foundation used to generate the graphs and charts. Readings for a given resource are typically spread across several tables:

- readings - Raw readings derived directly from devices. Data is periodically purged from this table.

- readings_rollups/hourly - Data from the readings table is rolled up into this table at hourly intervals.

- readings_rollups/daily - Data from the readings table is rolled up into this table at daily intervals.

- readings_rollups/monthly - Data from the readings table is rolled up into this table at monthly intervals.

As a convenience, in some cases a resource provides a shortcut method for returning readings. For example, the outlet resource:

```
GET    /api/v2/outlets/:outlet_id/readings
GET
/api/v2/outlets/:outlet_id/readings_rollups/hourly
GET
/api/v2/outlets/:outlet_id/readings_rollups/daily
GET
/api/v2/outlets/:outlet_id/readings_rollups/monthly
```

These methods return the same results as these:

```
GET    /api/v2/outlet_readings
GET    /api/v2/outlet_readings_rollups/hourly
GET    /api/v2/outlet_readings_rollups/daily
GET    /api/v2/outlet_readings_rollups/monthly
```

However, in the first case, the readings that are returned are only for the outlet specified by the outlet_id parameter. In the latter case, readings for multiple outlets are returned.

▶ **Route Reference**

```
GET  /api/v2/sensor_readings
GET  /api/v2/sensor_readings_rollups
GET  /api/v2/sensor_readings_rollups/hourly
GET  /api/v2/sensor_readings_rollups/daily
GET  /api/v2/sensor_readings_rollups/monthly
GET  /api/v2/sensors
GET  /api/v2/sensors/:id
GET  /api/v2/sensors/:sensor_id/events
GET  /api/v2/sensors/:sensor_id/readings
GET /api/v2/sensors/:sensor_id/readings_rollups [DEPRECATED]
GET  /api/v2/sensors/:sensor_id/readings_rollups/hourly
GET  /api/v2/sensors/:sensor_id/readings_rollups/daily
GET  /api/v2/sensors/:sensor_id/readings_rollups/monthly
```

```
GET   /api/v2/outlet_readings
GET  /api/v2/outlet_readings_rollups [DEPRECATED]
GET  /api/v2/outlet_readings_rollups/hourly
GET  /api/v2/outlet_readings_rollups/daily
GET  /api/v2/outlet_readings_rollups/monthly

GET  /api/v2/circuit_breaker_readings
GET  /api/v2/circuit_breaker_readings_rollups/ [DEPRECATED]
GET  /api/v2/circuit_breaker_readings_rollups/hourly
GET  /api/v2/circuit_breaker_readings_rollups/daily
GET  /api/v2/circuit_breaker_readings_rollups/monthly

GET   /api/v2/inlet_readings
GET   /api/v2/inlet_readings_rollups/hourly
GET   /api/v2/inlet_readings_rollups/daily
GET   /api/v2/inlet_readings_rollups/monthly
GET   /api/v2/inlet_pole_readings
GET   /api/v2/inlet_pole_readings_rollups/hourly
GET   /api/v2/inlet_pole_readings_rollups/daily
GET   /api/v2/inlet_pole_readings_rollups/monthly
```

Power Control

You can perform power control using the REST API.

Power control is supported for the following operations:

- Power control of EDM entities, such as racks
  - EDM entities not supported: data centers and floors

45

- Device groups
- Outlets

► **Route Reference**

```
# device groups
POST   /api/v2/device_groups/power_control

# outlets
POST   /api/v2/outlets/power_control

# EDM entities
POST   /api/v2/rooms/:id/power_control
POST   /api/v2/rows/:id/power_control
POST   /api/v2/aisles/:id/power_control
POST   /api/v2/racks/:id/power_control
POST   /api/v2/devices/:id/power_control
```

The following routes return a 400 Bad Request with an error message, rather than a 404 Not Found. Power control to data centers and floors is not supported:

```
POST   /api/v2/data_centers/:id/power_control
POST   /api/v2/floors/:id/power_control
```

► **Request**

Specify the power control operation in the request body:

```
 POST /api/v2/device_groups/power_control
 {
   "state": "ON",    // or "OFF" or "CYCLE", depending on what
the resource supports
   // ...
 }
```

For outlets and device groups, specify the EDM entity IDs as part of the request body:

```
 # outlets
 POST /api/v2/outlets/power_control
 {
   "state": "ON",
   "outlets": [1, 2, 3]
 }
```

```
# device groups
POST /api/v2/device_groups/power_control
{
  "state": "ON",
  "device_groups": [1, 2, 3]
}
```

Power control on EDM entities is supported for a single entity at a time, so rather than specifying the entity IDs in the request body, it's part of the URL. In this case, the power control operation is still part of the request body:

```
# power control on rack-3
POST /api/v2/racks/3/power_control
```

▶ **Response**

All of these operations return a job on success or an error message on failure.

▶ **Permissions:**

The API respects all the same administrative settings and permissions as in the Power IQ web client.

* Global power control
* (EDM node) power control
* Per-node permissions
* Graceful shutdown

Misc

These are additional resources and actions that don't fit inside the other functional areas.

▶ **Route Reference**

```
GET    /api/v2/system_info
```

GET /api/v2/system_info

Returns configuration information regarding the Power IQ system, including version number, poller settings, and more. There is no way to write system configuration information through the API. The Power IQ user interface must be used to configure the system.

Addendum A: Resource Reference

The following tables provide information on all resources available in Power IQ, with contextual information needed to understand how these resources can and cannot be used in the API. For details on actions that can be performed on these resources, and when these resources should be used in a request or when it appears in a response, see the API section of this document.

Modules

- *Aisle* (on page 49)
- *AssetStrip* (on page 50)
- *CircuitBreakerReadings* (on page 53)
- *CircuitBreakerReadings*Rollup* (on page 55)
- *Configuration* (on page 56)
- *DataCenter* (on page 63)
- *DcRack* (see "*Rack*" on page 107)
- *Device* (on page 66)
- *Event* (on page 68)
- *Floor* (on page 71)
- *InletPoleReading* (on page 75)
- *InletPoleReading*Rollup* (on page 78)
- *InletReading* (on page 81)
- *InletReading*Rollup* (on page 84)
- *Job* (on page 88)
- *JobMessage* (on page 90)
- *Licensing* (on page 91)
- *Outlet* (on page 92)
- *OutletReadings* (on page 94)
- *OutletReadings*Rollup* (on page 98)
- *Pdu* (on page 102)
- *Rack* (on page 107)
- *RackUnit* (on page 108)
- *Room* (on page 110)
- *Row* (on page 111)
- *Sensor* (on page 112)
- *SensorReadings* (on page 114)
- *SensorReadings*Rollups* (on page 115)
- *SystemInfo* (on page 117)

Aisle

Physical aisle of a data center. Part of the data center hierarchy that can be used to model your site.

► **Example:**

```
{"aisle" : {
 "id" : 2,
 "name" :"Aisle 2",
 "external_key" :"Aisle -- 2",
 "capacity" :1.0
 }}
```

Attribute Details

► **capacity**

User-defined power capacity in kW

- Type: Double
- Sample Values: 1.0

► **external_key**

A code used to uniquely identify this resource

- Type: String

► **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

► **name**

The name of the resource

- Type: String

AssetStrip

Asset management strips attached to PDUs or EMXs that are being managed by Power IQ.

Note: An EMX is treated the same as a PDU.

```
{"asset_strip":{
 "id":1,
 "pdu_id":27,
 "name":"Strip of Bacon",
 "state":"available",
 "created_at":"2011/10/07 14:50:01 +0000",
 "updated_at":"2011/10/07 14:50:01 +0000",
 "ordinal":1,
 "default_connected_led_color":"ff0000",
 "default_disconnected_led_color":"0000ff" }}
```

Attribute Details

▶ **created_at** (readonly)

The date and time this resource was created.

Note: The timezone for all date and time fields is always UTC (+0000), which reports the time on the Power IQ, not the time of your client. Time granularity is returned in seconds but internally stored to the usec. Be aware of this when performing searches against particular times.

- Type: String
- Sample Values: "2011/10/07 14:50:00 +0000"

▶ **default_connected_led_color**

The color of a RackUnit LED connected to the AssetStrip, in hexadecimal RGB.

- Type: String
- Sample Values: "FF0000"

▶ **default_disconnected_led_color**

The color of a RackUnit LED disconnected from the AssetStrip, in hexadecimal RGB.

- Type: String
- Sample Values: "CECECE"

EXE Raritan.

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **name**

The name of the resource

- Type: String

▶ **numbering_offset** (readonly)

The offset that the rack unit positions within this AssetStrip use at the beginning.

Note: An EMX is treated the same as a PDU.

- Type: Integer

▶ **numbering_scheme** (readonly)

The numbering scheme of rack units within the AssetStrip.

- Type: String
- Sample Values: "top_down", "bottom_up"

▶ **ordinal** (readonly)

Note: An EMX is treated the same as a PDU

The position within the PDU that this AssetStrip resource occupies

- Type: Integer

▶ **orientation** (readonly)

The orientation of the AssetStrip.

- Type: String
- Sample Values: "top_connector", "bottom_connector"

▶ **pdu_id** (readonly)

Note: An EMX is treated the same as a PDU

The ID of the PDU that this resource is associated with

- Type: Integer

▶ **state** (readonly)

The state of the AssetStrip.

- Type: String

- Sample Values: "ok","upgrading","unavailable","unsupported"

▶ **updated_at** (readonly)

- Type: String

- Sample Values: "2011/10/07 14:50:01 +0000"

Blade_Slot

Blade slots are analogous to rack units, except they attach to vertically oriented blade servers rather than horizontally oriented rack-mounted servers.

▶ **Examples**

```
{
  "blade_slot": {
    "id": 1,
    "rack_unit_id": 64,
    "ordinal": 1,
    "tag_id": "",
    "created_at": "2012/09/17 18:52:44 +0000",
    "updated_at": "2012/09/17 18:52:44 +0000"
  }
}
```

Attribute Details

► **id**

The unique identifier for the blade slot.

► **rack_unit_id**

The unique identifier for the rack_unit object through which the blade is connected to the asset strip.

► **ordinal**

The displayed number of the blade slot. The slot's unique number within the blade.

► **tag_id**

The asset tag id string of the asset tag connected to the blade slot.   If none this is empty.

► **created_at**

The timestamp when this blade slot was created.

► **updated_at**

The timestamp when this blade slot was last updated.

CircuitBreakerReadings

The CircuitBreakerReadings resource shows the power data collected from circuit breakers on the PDU. This resource does not contain any data for PDUs that do not have circuit breakers. A data record is added for each circuit breaker polled. This data is summarized hourly in the CircuitBreakerReadingsRollup resource.

Note: Circuit breaker readings are periodically purged

► **Examples:**
```
{"circuit_breaker_reading" : {
 "id":3538,
 "circuit_breaker_id":1,
 "reading_time":"2011/10/17 15:24:49 +0000",
 "current":0.0,
 "unutilized_capacity":20.0,
 "pdu_id":15,
 "max_current":null,
 "min_current":null}}
```

Attribute Details

▶ **circuit_breaker_id** (readonly)

The ID of the CircuitBreaker that this resource is associated with

- Type: Integer

▶ **current**

The current in amps

- Type: Float

▶ **current_amps** (readonly) DEPRECATED **Use #current**

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **max_current_amps** (readonly) DEPRECATED **Use #max_current**

The max_current_amps field

- Type: Float

▶ **min_current_amps** (readonly) DEPRECATED **Use #min_current**

The min_current_amps field

- Type: Float

▶ **pdu_id** (readonly)

Note: An EMX is treated the same as a PDU

The ID of the PDU that this resource is associated with

- Type: Integer

▶ **reading_time** (readonly)

The date and time the reading was collected

- Type: String
- Sample Values: "2011/10/07 14:50:01 +0000"

▶ **unutilized_capacity** (readonly)

The unutilized_capacity in amps.

- Type: Float)

CircuitBreakerReadings*Rollup

The CircuitBreakerReadings*Rollup resource summarizes the outlet readings power data over the rollup interval. The "*" may be replaced with the string Hourly, Daily, or Monthly. For example, CircuitBreakerReadingsHourlyRollup.

Raw data is rolled up every hour, hourly roll-ups are in turn rolled up once a day, and daily roll-ups are rolled up once a month.

▶ **Examples:**

```
{
  "circuit_breaker_readings_hourly_rollup": {
    "id": 1,
    "circuit_breaker_id": 13,
    "reading_time": "2011/10/07 14:00:00 +0000",
    "min_current": 0.8,
    "max_current": 0.8,
    "average_current": 0.8,
    "min_unutilized_capacity": 19.2,
    "max_unutilized_capacity": 19.2,
    "average_unutilized_capacity": 19.2,
    "pdu_id": 26
  }
}
```

Attribute Details

▶ **average_current** (readonly)

The average_current field

- Type: Float

▶ **average_unutilized_capacity** (readonly)

The average_unutilized_capacity field

- Type: Float

▶ **circuit_breaker_id** (readonly)

The ID of the CircuitBreaker that this resource is associated with

- Type: Integer

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **max_current** (readonly)

The max_current field

- Type: Float

▶ **max_unutilized_capacity** (readonly)

The max_unutilized_capacity field

- Type: Float

▶ **min_current** (readonly)

The min_current field in amps

- Type: Float

▶ **min_unutilized_capacity** (readonly)

The min_unutilized_capacity field

- Type: Float

▶ **pdu_id** (readonly)

Note: An EMX is treated the same as a PDU

The ID of the PDU that this resource is associated with

- Type: Integer

▶ **reading_time** (readonly)

The date and time the reading was collected

- Type: String
- Sample Values: "2011/10/07 14:50:01 +0000"

Configuration

Attribute Details

▶ **allow_html_portlets** (readonly)

The allow_html_portlets field

- Type: Boolean

▶ **browser_session_polling_interval** (readonly)

The browser_session_polling_interval field

- Type: Integer

▶ **canonical_domain** (readonly)

The canonical_domain field

- Type:String

▶ **currency** (readonly)

The currency field

- Type: String

▶ **data_center_custom_field_1** (readonly)

The data_center_custom_field_1 field

- Type: String

▶ **data_center_custom_field_2** (readonly)

The data_center_custom_field_2 field

- Type:String

▶ **device_custom_field_1** (readonly)

The device_custom_field_1 field

- Type:String

▶ **device_custom_field_2** (readonly)

The device_custom_field_2 field

- Type: String

▶ **enable_browser_power_control** (readonly)

The enable_browser_power_control field

- Type: Boolean

▶ **enable_custom_logo** (readonly)

The enable_custom_logo field

- Type:Boolean

▶ **enable_event_notifications** (readonly)

The enable_event_notifications field

- Type:Boolean

▶ **enable_power_control** (readonly)

The enable_power_control field

- Type: Boolean

▶ **enable_scheduled_power_control** (readonly)

The enable_scheduled_power_control field

- Type: Boolean

▶ **enable_web_api** (readonly)

The enable_web_api field

- Type: Boolean

▶ **enable_web_api_user** (readonly)

The enable_web_api_user field

- Type: Boolean

▶ **eula_accepted** (readonly)

The eula_accepted field

- Type: Boolean

▶ **from_email** (readonly)

The from_email field

- Type: String

▶ **login_body** (readonly)

The login_body field

- Type: String

▶ **login_header** (readonly)

The login_header field

- Type: String

▶ **ntp_enabled** (readonly)

The ntp_enabled field

- Type: Boolean

▶ **password_max_length** (readonly)

The password_max_length field

- Type: Integer

▶ **password_min_length** (readonly)

The password_min_length field

- Type: Integer

▶ **password_requires_one_lowercase** (readonly)

The password_requires_one_lowercase field

- Type: Boolean

▶ **password_requires_one_numeric** (readonly)

The password_requires_one_numeric field

- Type:Boolean

▶ **password_requires_one_special** (readonly)

The password_requires_one_special field

- Type: Boolean

▶ **password_requires_one_uppercase** (readonly)

The password_requires_one_uppercase field

- Type: Boolean

▶ **pdu_custom_field_1** (readonly)

The pdu_custom_field_1 field

- Type: String

▶ **pdu_custom_field_2** (readonly)

The pdu_custom_field_2 field

- Type:String

▶ **pdu_label** (readonly)

The pdu_label field

- Type: String

▶ **px_minimum_version** (readonly)

The px_minimum_version field

- Type: String

▶ **remote_storage_directory** (readonly)

The remote_storage_directory field

- Type: String

▶ **remote_storage_enabled** (readonly)

The remote_storage_enabled field

- Type: Boolean

▶ **remote_storage_ftp_user** (readonly)

The remote_storage_ftp_user field

- Type: String

▶ **remote_storage_host** (readonly)

The remote_storage_host field

- Type: String

▶ **remote_storage_port** (readonly)

The remote_storage_port field

- Type: Integer

▶ **remote_storage_protocol** (readonly)

The remote_storage_protocol field

- Type: String

▶ **remote_storage_push_backup** (readonly)

The remote_storage_push_backup field

- Type: Boolean

▶ **remote_storage_push_csv** (readonly)

The remote_storage_push_csv field

- Type: Boolean

▶ **remote_storage_s3_ssl_only** (readonly)

The remote_storage_s3_ssl_only field

- Type: Boolean

▶ **require_power_control_audit_message** (readonly)

The require_power_control_audit_message field

- Type: Boolean

▶ **rss_enabled** (readonly)

The rss_enabled field

- Type: Boolean

▶ **rss_extra_text** (readonly)

The rss_extra_text field

- Type: String

▶ **session_timeout** (readonly)

The session_timeout field

- Type: Integer

▶ **site_locale** (readonly)

The site_locale field

- Type: String

▶ **smtp_auth_type** (readonly)

The smtp_auth_type field

- Type: String

▶ **smtp_encryption_method** (readonly)

The smtp_encryption_method field

- Type: String

▶ **smtp_port** (readonly)

The smtp_port field

- Type: Integer

▶ **smtp_server** (readonly)

The smtp_server field

- Type: String

▶ **smtp_username** (readonly)

The smtp_username field

- Type: String

▶ **snmp_version** (readonly)

The snmp_version field

- Type: String

▶ **sso_enabled** (readonly)

The sso_enabled field

- Type: String

▶ **temperature_unit** (readonly)

The temperature_unit field

- Type: String

▶ **time_zone** (readonly)

The time_zone field

- Type: String

▶ **time_zone_offset** (readonly)

The time_zone_offset field

- Type: String

DataCenter

Physical data center. Part of the data center hierarchy that can be used to model your site. A data center is always the topmost element in the data center hierarchy.

▶ **Examples:**

```
{"data_center" : {
 "id" : 2,
 "name" :"My Data Center",
 "company_name" :"MyCompany",
 "contact_name" :"MyName",
 "contact_phone" : "999-999-9999",
 "contact_email" : "MyName@MyCompany.co",
 "city" : "Raleigh",
 "state" : "NC",
 "country" : "USA",
 "peak_kwh_rate" : 0.1,
 "off_peak_kwh_rate" : 0.06,
 "peak_begin" : 7.0,
 "peak_end" : 19.0,
 "co2_factor" : 0.6,
 "cooling_factor" : 1.0,
 "custom_field_1" : "field1",
 "custom_field_2" :"field2",
 "external_key" :"Data Center -- 2",
 "capacity" :1.0,
 "cooling_savings" : 7.0
}}
```

Attribute Details

▶ **capacity**

User-defined power capacity in kW

- Type: Double
- Sample Values: 1.0

▶ **city**

City location for the DataCenter

- Type: String

▶ **co2_factor**

User-defined CO2 computational factor

- Type: Double

▶ **company_name**

Name of the company that corresponds to the DataCenter

- Type: String

▶ **contact_email**

Email of the person to contact for the DataCenter

- Type: String

▶ **contact_name**

Name of the person to contact for the DataCenter

- Type: String

▶ **contact_phone**

Phone of the person to contact for the DataCenter

- Type: String

▶ **cooling_factor**

User-defined cooling computational factor

- Type: Double

▶ **cooling_savings**

User-defined savings % per degrees C

- Type: **(**Double**)**
- Sample Values: 7

▶ **country**

Country location for the DataCenter

- Type: **(**String**)**

▶ **custom_field_1**

User-defined customizable field

- Type: **(**String**)**

▶ **custom_field_2**

User-defined customizable field

- Type: String

▶ **external_key**

A code used to uniquely identify this resource

- Type: String

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **name**

The name of the resource

- Type: String

▶ **off_peak_kwh_rate**

User-defined energy cost per Kilowatt Hour during off-peak hours

- Type: Double

▶ **peak_begin**

User-defined time of day peak hours begin

- Type: Double
- Sample Values: 19.5 (corresponding to 19:30 hours)

▶ **peak_end**

User-defined time of day peak hours ends

- Type: Double
- Sample Values: 19.5 (corresponding to 19:30 hours)

▶ **peak_kwh_rate**

User-defined energy cost per Kilowatt Hour during peak hours

- Type: Double

▶ **state**

State location for the DataCenter

- Type: String

Device

Physical device in a data center. Part of the data center hierarchy that can be used to model your site. Unlike other data center hierarchy resources, devices can have outlets associated with them. Devices can also be associated with AssetStrips through RackUnits.

▶ **Examples:**

```
{"device":{
 "id":143,
 "name":"Device for PDU 192.168.43.42 outlet not used",
 "customer":"Unknown", "device_type":"Default Generated
Device",
 "power_rating":100,
 "decommissioned":true,
 "custom_field_1":"field1",
 "custom_field_2":"field2",
 "external_key":"IT Device -- 143",
 "ip_address":"192.168.43.122",
 "asset_tag_id":""
 }}
```

Attribute Details

▶ **asset_tag_id** (readonly)

An internally generated unique identifier that is used to associate a device with a RackUnit. The value comes directly from the RackUnit asset strip.

- Type: String

▶ **custom_field_1**

User-defined customizable field

- Type: String

▶ **custom_field_2**

User-defined customizable field

- Type: String

▶ **customer**

Customer that the Device belongs to

- Type: String

▶ **decommissioned**

User-defined flag to indicate whether this device is decommissioned

- ▪ Type: Boolean

▶ **device_type**

User-defined device type

- ▪ Type: String

▶ **external_key**

A code used to uniquely identify this resource

- ▪ Type: String

▶ **id** (readonly)

An automatically generated ID for this resource

- ▪ Type: Integer

▶ **ip_address**

IP address for the Device

- ▪ Type: String

▶ **name**

The name of the resource

- ▪ Type: String

▶ **power_rating**

User-defined power in watts that this device is rated for

- ▪ Type: Integer

Device Groups

Groups of physical devices in a data center, created for purposes of organization, and for power control of devices in the group.

▶ **Examples:**

```
# device groups
POST /api/v2/device_groups/power_control
{
  "state": "ON",
  "device_groups": [1, 2, 3]
}
```

Attribute Details

Device group attributes are not exposed through the REST API. The API can only control power to device groups.

Event

Events generated typically by PDU SNMP traps forwarded to Power IQ. Events can also be generated internally by Power IQ.

▶ **Examples:**

```
{"event":{
 "id":4,
 "event_config_id":62,
 "source":2,
 "created_at":"2011/10/07 16:02:46 +0000",
 "pdu_id":15, "pdu_outlet_id":null,
 "pdu_circuitbreaker_id":null,
 "sensor_id":null,
 "trap_oid":null,
 "cleared_by":3,
 "cleared_at":"2011/10/07 19:29:26 +0000",
 "clearing_event_id":null,
 "clearing_user_id":null,
 "notification_status":6,
 "asset_strip_id":null,
 "rack_unit_id":null,
 "params":[
 {
 "key":"test",
 "value":"test"
 },
 {
 "key":"timestamp",
 "value":"1317999602806"
 }
 ]
 }}
```

Attribute Details

▶ **asset_strip_id** (readonly)

The ID of the AssetStrip that this resource is associated with

- Type: Integer

▶ **blade_slot_id (**readonly)

The ID of the associated BladeSlot.

- Type: Integer

▶ **cleared_at** (readonly)

The date and time the event was cleared

- Type: String

▶ **cleared_by** (readonly)

The source from which the event was cleared by

- Type: Integer
- Sample Values: 1 = Event, 2 = User, 3 = Trap

▶ **clearing_event_id** (readonly)

The ID of the Event that this event was cleared by

- Type: Integer

▶ **clearing_user_id** (readonly)

Note: User resources cannot be retrieved by the API

The ID of the User that cleared this event

- Type: Integer

▶ **created_at** (readonly)

The date and time this resource was created

- Type: String
- Sample Values: "2011/10/07 14:50:00 +0000"

▶ **event_config_id** (readonly)

The EventConfig that this resource is associated with. The EventConfig can be used to identify the type of event that occurred.

- Type: Integer

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

► **notification_status** (readonly)

The current state or results of the notification that was sent regarding this event

- Type: Integer

- Sample Values: 1 = Unsent, 2 = Sent Active, 3 = Sent Cleared, 4 = Failed Active, 5 = Failed Cleared, 6 = Processed

► **outlet_id** (readonly)

The ID of the associated Outlet.

- Type: Integer

► **params** (readonly)

Multi-varied parameters associated with the event

- Type: Mixed

- Sample Values: "key":"key1","value":"val1", "key":"key2","value":"val2"

► **pdu_circuitbreaker_id** (readonly) DEPRECATED **Use #circuit_breaker_id**

The ID of the CircuitBreaker that this resource is associated with

- Type: Integer

► **pdu_id** (readonly)

Note: An EMX is treated the same as a PDU

The ID of the PDU that this resource is associated with

- Type: Integer

► **pdu_outlet_id** (readonly) DEPRECATED **Use #outlet_id**

The ID of the Outlet that this resource is associated with

- Type: Integer

► **rack_unit_id** (readonly)

The ID of the RackUnit that this resource is associated with

- Type: Integer

▶ **sensor_id** (readonly)

The ID of the sensor that this resource is associated with

- Type: Integer

▶ **source** (readonly)

The source identifies where the event came from

- Type: Integer
- Sample Values: 1 = SNMP Trap, 2 = Generated internally

▶ **trap_oid** (readonly)

The SNMP OID of the trap

- Type: String

Floor

Physical floor in a data center. Part of the data center hierarchy that can be used to model your site.

▶ **Examples:**

```
{"floor" : {
 "id" : 2,
 "name" :"Floor 2",
 "external_key" :"Floor -- 2",
 "capacity" :1.0
}}
```

Attribute Details

▶ **capacity**

User-defined power capacity in kW

- Type: Double
- Sample Values: 1.0

▶ **external_key**

A code used to uniquely identify this resource

- Type: String

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **name**

The name of the resource

- Type: String

Inlet

The Inlet resource represents a single inlet on a PDU. PDUs may have one or more inlets, and inlets may contain one or more inlet poles. Generally, inlets belonging to single-phase PDUs will have one associated inlet pole, and inlets belonging to three-phase PDUs will have three associated inlet poles.

▶ **Examples:**

```
{
  "inlet": {
    "id": 1,
    "pdu_id": 4,
    "ordinal": 1,
    "rated_amps": 15,
    "reading": {
      "id": 44159,
      "pdu_id": 4,
      "inlet_id": 1,
      "reading_time": "2013/02/04 13:58:34 -0500",
      "voltage": null,
      "min_voltage": null,
      "max_voltage": null,
      "current": 0,
      "min_current": null,
      "max_current": null,
      "unutilized_capacity": 15,
      "min_unutilized_capacity": null,
      "max_unutilized_capacity": null,
      "power_factor": 1,
      "min_power_factor": null,
      "max_power_factor": null,
      "active_power": 0,
      "min_active_power": null,
      "max_active_power": null,
      "apparent_power": 0,
      "min_apparent_power": null,
      "max_apparent_power": null,
      "volt_amp_hour": null,
      "watt_hour": null,
      "inlet_ordinal": 1
    }
  }
}
```

Attribute Details

▶ **id** (readonly)

An automatically generated ID for this resource.

- Type: Integer

▶ **ordinal** (readonly)

The ordinal of the inlet on the PDU (i.e., inlet 1, inlet 2, etc.).

- Type: Integer

▶ **pdu_id** (readonly)

Note: An EMX is treated the same as a PDU.

The ID of the associated PDU.

- Type: Integer

▶ **rated_amps** (readonly)

Rated Amps (A) on the inlet.

- Type: Integer

▶ **reading** (readonly)

The latest reading for this inlet. See ***InletReading*** (on page 81).

Inlet Pole

The InletPole resource represents a single wire within an inlet on a PDU.

▶ **Examples:**

```
{
  "inlet_pole": {
    "id": 18,
    "inlet_id": 11,
    "pdu_id": 14,
    "ordinal": 2,
    "reading": {
      "id": 68099,
      "reading_time": "2013/02/04 14:17:12 -0500",
      "current": 2.971,
      "unutilized_capacity": 29.029,
      "pdu_id": 14,
      "max_current": null,
      "min_current": null,
      "inlet_pole_id": 18,
      "voltage": 204.96,
      "min_voltage": null,
      "max_voltage": null,
      "min_unutilized_capacity": null,
      "max_unutilized_capacity": null,
      "inlet_id": 11,
      "inlet_ordinal": 3,
      "inlet_pole_ordinal": 2
    }
  }
}
```

Attribute Details

▶ **id** (readonly)

An automatically generated ID for this resource.

- Type: Integer

▶ **inlet_id** (readonly)

The ID of the associated Inlet.

- Type: Integer

▶ **ordinal** (readonly)

The ordinal of the inlet pole on the PDU inlet (i.e., pole 1, pole 2, etc.).

- Type: Integer

▶ **pdu_id** (readonly)

Note: An EMX is treated the same as a PDU.

The ID of the associated PDU.

- Type: Integer

▶ **reading** (readonly)

The latest reading for this inlet pole. See ***InletPoleReading*** (on page 75).

InletPoleReading

The InletPoleReading resource shows the power data collected from PDU current-carrying poles. A data record is added for each pole polled. Single-phase PDUs have 1 pole. Three-phase PDUs have 3 poles. This data is summarized hourly in an InletPoleReadingRollup resource.

Note: Inlet pole readings are periodically purged.

▶ **Examples:**

```
{
  "inlet_pole_reading": {
    "id": 576,
    "pdu_id": 9,
    "inlet_id": 3,
    "inlet_ordinal": 1,
    "inlet_pole_id": 1,
    "inlet_pole_ordinal": 1,
    "reading_time": "2011/10/21 13:51:12 -0400",
    "current": 1.0,
    "max_current": 1.0,
    "min_current": 1.0,
    "unutilized_capacity": 23.0,
    "max_unutilized_capacity": 23.0,
    "min_unutilized_capacity": 23.0,
    "voltage": 207.0,
    "max_voltage": 207.0,
    "min_voltage": 207.0
  }
}
```

Attribute Details

▶ **current** (readonly)

The current in amps.

- Type: Float

▶ **current_amps** (readonly) DEPRECATED **Use #current.**

The amp reading for the line

- Type: Double

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **inlet_id** (readonly)

The ID of the associated Inlet.

- Type: Integer

▶ **inlet_ordinal** (readonly)

The ordinal of the inlet on the PDU (i.e., inlet 1, inlet 2, etc.).

- Type: Integer

▶ **inlet_pole_id** (readonly)

The ID of the associated InletPole.

- Type: Integer

▶ **inlet_pole_ordinal** (readonly)

The ordinal of the inlet pole on the PDU inlet (i.e., pole 1, pole 2, etc.).

- Type: Integer

▶ **max_current** (readonly)

The max_current in amps.

- Type: Float

▶ **max_unutilized_capacity** (readonly)

The max_unutilized_capacity in amps.

- Type: Float

▶ **max_voltage** (readonly)

The max_voltage in volts.

- Type: Float

▶ **min_current** (readonly)

The min_current in amps.

- Type: Float

▶ **min_unutilized_capacity** (readonly)

The min_unutilized_capacity in amps.

- Type: Float

▶ **min_voltage** (readonly)

The min_voltage in volts.

- Type: Float

▶ **pdu_id** (readonly)

Note: An EMX is treated the same as a PDU

The ID of the PDU that this resource is associated with

- Type: Integer

▶ **reading_time** (readonly)

The date and time the reading was collected

- Type: String
- Sample Values: "2011/10/07 14:50:01 +0000"

▶ **unutilized_capacity** (readonly)

Unutilized capacity (Amps)

- Type: Float

▶ **voltage**(readonly)

The voltage in volts.

- Type: Float

InletPoleReading*Rollup

Note: When data is rolled up, shorter interval roll-up entries in the InletPoleReadingRollup are purged.

The InletPoleReading*Rollup resources summarize the inlet pole reading power data over the rollup interval. The "*" may be replaced with the string Hourly, Daily, or Monthly. For example, InletPoleReadingHourlyRollup.

Raw data is rolled up every hour, hourly roll-ups are in turn rolled up once a day, and daily roll-ups are rolled up once a month.

▶ **Examples:**

```json
{
  "inlet_pole_reading_hourly_rollups": [
    {
      "id": 2,
      "max_voltage": 118,
      "pdu_id": 41,
      "average_voltage": 116.351,
      "reading_time": "2012/09/12 13:00:00 -0400",
      "min_current": 0.749,
      "average_unutilized_capacity": 10.8366,
      "max_unutilized_capacity": 11.251,
      "inlet_pole_id": 38,
      "min_voltage": 115,
      "average_current": 1.16337,
      "max_current": 1.702,
      "min_unutilized_capacity": 10.298
    }
  ]
}

{
  "inlet_pole_reading_daily_rollups": [
    {
      "id": 7,
      "max_voltage": 119,
      "pdu_id": 41,
      "average_voltage": 116.507,
      "reading_time": "2012/09/12",
      "min_current": 1.139,
      "average_unutilized_capacity": 10.8081,
      "max_unutilized_capacity": 10.861,
      "inlet_pole_id": 38,
      "min_voltage": 115,
      "average_current": 1.19194,
      "max_current": 1.355,
      "min_unutilized_capacity": 10.645
    }
```

```
    ]
}
```

Attribute Details

▶ **average_current** (readonly)

Average current (Amps) reading on the line during rollup interval

- Type: Double

▶ **average_unutilized_capacity** (readonly)

Average unutilized capacity (Amps) on the line during rollup interval

- Type: Double

▶ **average_voltage** (readonly)

The average_voltage in volts.

- Type: Float

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **inlet_pole_id readonly**

The ID of the associated InletPole.

- Type: Integer

▶ **max_current** (readonly)

The max_current in amps.

- Type: Float

▶ **max_unutilized_capacity** (readonly)

The max_unutilized_capacity in amps.

- Type: Float

▶ **max_voltage** (readonly)

The max_voltage in volts.

- Type: Float

▶ **min_current** (readonly)

The min_current in amps.

- Type: Float

▶ **min_unutilized_capacity** (readonly)

The min_unutilized_capacity in amps.

- Type: Float

▶ **min_voltage** (readonly)

The min_voltage in volts.

- Type: Float

▶ **pdu_id** (readonly)

Note: An EMX is treated the same as a PDU

The ID of the PDU that this resource is associated with

- Type: Integer

▶ **reading_time** (readonly)

The date and time the reading was collected

- Type: String
- Sample Values: "2011/10/07 14:50:01 +0000"

InletReading

The InletReading view shows the raw power data collected from PDUs. A data record is added for each inlet of each PDU polled. Inlets are also referred to as infeed, or line in. This data is summarized hourly in InletReadingHourlyRollup, and the readings in this view are purged.

▶ **Examples:**

```
{
  "inlet_reading": {
    "inlet_id": 16,
    "inlet_ordinal": 1,
    "volt_amp_hour": null,
    "min_unutilized_capacity": null,
    "max_power_factor": null,
    "min_apparent_power": null,
    "voltage": 118.5,
    "active_power": 0,
    "min_voltage": null,
    "pdu_id": 27,
    "max_current": null,
    "current": 0,
    "min_current": null,
    "max_apparent_power": null,
    "unutilized_capacity": 12,
    "watt_hour": null,
    "power_factor": 0,
    "id": 31447,
    "max_unutilized_capacity": null,
    "max_voltage": null,
    "min_power_factor": null,
    "max_active_power": null,
    "apparent_power": 0,
    "reading_time": "2012/10/25 14:09:39 -0400",
    "min_active_power": null
  }
}
```

Attribute Details

▶ **active_power** (readonly)

Active Power drawn by the PDU

- Type: Double

▶ **apparent_power** (readonly)

Apparent Power drawn by the PDU

- Type: Double

▶ **current** (readonly)

The current in amps.

- Type: Float

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **inlet_ordinal** (readonly)

The ordinal of the inlet on the PDU (i.e., inlet 1, inlet 2, etc.).

- Type: Integer

▶ **inlet_id** (readonly)

The ID of the associated Inlet.

- Type: Integer

▶ **max_active_power** (readonly)

The max_active_power in watts.

- Type: Float

▶ **max_apparent_power** (readonly)

The max_apparent_power in volt-amps.

- Type: Float

▶ **max_current** (readonly)

The max_current in amps.

- Type: Float

▶ **max_power_factor** (readonly)

Maximum power factor on the inlet.

- Type: Double

▶ **max_unutilized_capacity** (readonly)

The max_unutilized_capacity in amps.

- Type: Float

▶ **max_voltage** (readonly)

The max_voltage in volts.

- Type: Float

▶ **min_current** (readonly)

The min_current in amps.

- Type: Float

▶ **min_active_power** (readonly)

The min_active_power in watts.

- Type: Float

▶ **min_apparent_power** (readonly)

The min_apparent_power in volt-amps.

- Type: Float

▶ **min_unutilized_capacity** (readonly)

The min_unutilized_capacity in amps.

- Type: Float

▶ **min_voltage** (readonly)

The min_voltage in volts.

- Type: Float

▶ **pdu_id** (readonly)

Note: An EMX is treated the same as a PDU

- The ID of the PDU that this resource is associated with
- Type: Integer

▶ **power_factor** (readonly)

Power factor on the inlet.

- Type: Double

▶ **reading_time** (readonly)

The date and time the reading was collected

- Type: String
- Sample Values: "2011/10/07 14:50:01 +0000"

▶ **unutilized_capacity** (readonly)

Unutilized capacity (Amps)

- Type: Float

▶ **volt_amp_hour** (readonly)

Total volt-amp-hours on the inlet.

- Type: Double

▶ **voltage** (readonly)

The voltage in volts.

- Type: Float

▶ **watt_hour** (readonly)

Total watt hours on the inlet.

- Type: Double

InletReading*Rollup

- InletReadingsHourlyRollup
- InletReadingsDailyRollup
- InletReadingsMonthlyRollup

The InletReadingsRollup resource summarizes the inlet reading power data over the roll-up interval. The "*" may be replaced with the string Hourly, Daily, or Monthly. For example, InletReadingHourlyRollup.

Raw data is rolled up every hour, hourly roll-ups are in turn rolled up once a day, and daily roll-ups are rolled up once a month.

▶ **Examples:**

```
{ "inlet_readings_hourly_rollups" : {
 "average_active_power" : 180.667,
 "average_apparent_power" : 239.333,
 "id" : 3,
 "max_active_power" : 181.0,
 "max_apparent_power" : 249.0,
 "min_active_power" : 180.0,
 "min_apparent_power" : 234.0,
 "pdu_id" : 9,
 "reading_time" : "2011/10/20 18:00:00 -0400",
 "rollup_interval" : 1,
 "watt_hour" : 1358360.0
}
```

Attribute Details

▶ **average_active_power** (readonly)

Average active power (Watts) reading during rollup interval

- Type: Double

▶ **average_apparent_power** (readonly)

Average apparent power (VA) reading during rollup interval

- Type: Double

▶ **average_current** (readonly)

The average_current field

- Type: Float

▶ **average_power_factor** (readonly)

Average power factor on the inlet.

- Type: Double

▶ **average_unutilized_capacity** (readonly)

The average_unutilized_capacity field

- Type: Float

▶ **average_voltage** (readonly)

The average_voltage in volts.

- Type: Float

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **inlet_id** (readonly)

The ID of the associated Inlet.

- Type: Integer

▶ **max_active_power** (readonly)

Maximum active power (Watts) reading during rollup interval

- Type: Double

▶ **max_apparent_power** (readonly)

Maximum apparent power (VA) reading during rollup interval

- Type: Double

▶ **max_current** (readonly)

The max_current in amps.

- Type: Float

▶ **max_power_factor** (readonly)

Maximum power factor on the inlet.

- Type: Double

▶ **max_unutilized_capacity** (readonly)

The max_unutilized_capacity in amps.

- Type: Float

▶ **max_voltage** (readonly)

The max_voltage in volts.

- Type: Float

▶ **min_current** (readonly)

The min_current in amps.

- Type: Float

▶ **min_active_power** (readonly)

Lowest active power (Watts) reading during rollup interval

- Type: Double

▶ **min_apparent_power** (readonly)

Lowest apparent power (VA) reading during rollup interval

- Type: Double

▶ **min_current** (readonly)

The min_current field in amps

- Type: Float

▶ **min_power_factor** (readonly)

Minimum power factor on the inlet.

- Type: Double

▶ **min_unutilized_capacity** (readonly)

The min_unutilized_capacity field

- Type: Float

▶ **min_voltage** (readonly)

The min_voltage in volts.

- Type: Float

▶ **pdu_id** (readonly)

Note: An EMX is treated the same as a PDU

The ID of the PDU that this resource is associated with

- Type: Integer

▶ **reading_time** (readonly)

The date and time the reading was collected

- Type: String
- Sample Values: "2011/10/07 14:50:01 +0000"

▶ **volt_amp_hour** (readonly)

Total volt-amp-hours on the inlet.

- Type: Double

▶ **watt_hour** (readonly)

Total watt hours on the inlet.

- Type: Double

Job

A Job is used to execute certain tasks in Power IQ. These tasks usually involve the execution of SNMP queries, or other heavy duty processing. A Job resource is returned by the REST API when a request is executed asynchronously. A Job will have one or more JobMessages, which describe the execution of the job in greater detail.

▶ **Examples:**

```
{"job":{
 "id":1,
 "user_id":1,
 "status":"COMPLETED",
 "description":null,
 "start_time":"2011/10/07 14:54:32 +0000",
 "end_time":"2011/10/07 14:54:33 +0000",
 "has_errors":false,
 "percent_complete":1.0,
 "completed":true,
 "last_message":"Discovered PDU 192.168.43.122, now
queuing for poll.",
 "error_count":0
}}
```

Attribute Details

▶ **completed** (readonly)

Indicates whether the job has finished or not

- Type: Boolean

▶ **end_time** (readonly)

The date and time the sub-task of the job that this JobMessage is related to was completed

- Type: String
- Sample Values: "2011/10/07 14:50:00 +0000"

▶ **error_count** (readonly)

The number of errors that occurred during execution of this Job

- Type: Integer

▶ **has_errors** (readonly)

Indicates whether the job has errors or not

- Type: Boolean

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **last_message** (readonly)

The text of the most recent JobMessage for the job

- Type: String

▶ **percent_complete** (readonly)

Indicates how much progress towards completion the job is as a percentage

- Type: Float

▶ **start_time** (readonly)

Note: The timezone for all date and time fields is always UTC (+0000). Time granularity is returned in seconds but internally stored to the usec. Be aware of this when performing searches against particular times.

The date and time the sub-task of the job that this JobMessage is related to was completed

- Type: String
- Sample Values: "2011/10/07 14:50:00 +0000"

▶ **status** (readonly)

The status of the Job

- Type: String
- Sample Values: ACTIVE = job is still running, COMPLETED = job has finished, ABORTED = job was aborted

▶ **user_id** (readonly)

The User that created the Job

- Type: Integer

JobMessage

A JobMessage describes detailed information about an executed Job.

▶ **Examples:**

```
{"job_message":{
 "id":1,
 "unit_of_work":0.5,
 "job_id":1,
 "level":"INFO",
 "trace":null,
 "start_time":"2011/10/07 14:54:33 +0000",
 "end_time":null,

"message_key":":magic.pdu_discovered_sysoid_no_proxy_
id",

"message_vars":"{\"sysoid\":\"1.3.6.1.4.1.13742.4\",\
"ip\":\"192.168.43.122\"}",
 "aborted":false,
 "message":"Discovering PDU 192.168.43.122 and found
SystemObjectID of 1.3.6.1.4.1.13742.4"
}}
```

Attribute Details

▶ **aborted** (readonly)

The sub-task of the job that this JobMessage is related to was aborted

- Type: String

▶ **end_time** (readonly)

The date and time the sub-task of the job that this JobMessage is related to was completed

- Type: String
- Sample Values: "2011/10/07 14:50:00 +0000"

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **job_id** (readonly)

The ID of the Job that this JobMessage is associated with

- Type: Integer

≡€ Raritan.

▶ **level** (readonly)

Indicates the severity of the message

- Type: String
- Sample Values: DEBUG, INFO, WARN, ERROR, FATAL, UNKNOWN

▶ **message** (readonly)

Human readable, localized job message text.

- Type: String

▶ **start_time** (readonly)

Note: The timezone for all date and time fields is always UTC (+0000). Time granularity is returned in seconds but internally stored to the usec. Be aware of this when performing searches against particular times.

The date and time the sub-task of the job that this JobMessage is related to was started

- Type: String
- Sample Values: "2011/10/07 14:50:00 +0000"

▶ **trace** (readonly)

Diagnostic stack trace if one is available; only present if there is an error in the job.

- Type: String

▶ **unit_of_work** (readonly)

The amount of work the sub-task of the job that this JobMessage is related to performed

- Type: String

Licensing

▶ **Examples:**
```
{ "licensing":{
 "device_limit":500,
 "customer_name":"Internal Use Only",
 "enable_events":false
}}
```

Attribute Details

▶ **customer_name** (readonly)

Customer name the license is under

- Type: String

▶ **device_limit** (readonly)

The maximum number of PDUs the current license allows Power IQ to manage

- Type: String

▶ **enable_events** (readonly)

If set to true, the license allows for collection of events

- Type: Boolean

Outlet

Outlets resources represent outlets associated with a PDU being managed by Power IQ.

▶ **Examples:**
```
{ "outlets" :
 { "device_id" : null,
 "id" : 17,
 "outlet_id" : 1,
 "outlet_name" : "Outlet_1",
 "pdu_id" : 9, "state" : "ON"
}}
```
Attribute Details

▶ **device_id**

The ID of the device this outlet is associated with.

- Type: Integer

▶ **ordinal** (readonly)

Outlet number on the pdu.

- Type: Integer

▶ **outlet_id** (readonly) DEPRECATED **Use #ordinal instead**

Outlet number on the PDU

- Type: Integer

▶ **outlet_name**

The name of the outlet

- Type: String

▶ **pdu_id** (readonly)

Note: An EMX is treated the same as a PDU

The ID of the PDU that this resource is associated with

- Type: Integer

▶ **rated_amps** (readonly)

The rated_amps in amps.

- Type: Float

▶ **state** (readonly)

State of the outlet

- Type: String
- Sample Values: ON, OFF

OutletReadings

The OutletReadings view shows the raw power data collected from PDU outlets. A data record is added for each outlet polled. This data is summarized hourly in a OutletReadings*Rollup view, and the outlet records in this view are purged.

▶ **Examples:**

```
{
  "outlet_reading": {
    "id": 5669,
    "pdu_id": 9,
    "outlet_id": 24,
    "reading_time": "2011/10/21 12:11:12 -0400",
    "active_power": 74.0,
    "max_active_power": 74.0,
    "min_active_power": 74.0,
    "apparent_power": 75.0,
    "max_apparent_power": 75.0,
    "min_apparent_power": 75.0,
    "current": 0.60,
    "max_current": 0.60,
    "min_current": 0.60,
    "current_amps": 0.60,
    "max_current_amps": 0.60,
    "min_current_amps": 0.60,
    "unutilized_capacity": 2.8,
    "max_unutilized_capacity": 2.8,
    "min_unutilized_capacity": 2.8,
    "watt_hour": 1593834,
    "voltage": 208.0,
    "max_voltage": 208.0,
    "min_voltage": 208.0
  }
}
```

Attribute Details

▶ **active_power** (readonly)

Active Power drawn by the outlet

- Type: Double

▶ **apparent_power** (readonly)

Apparent Power drawn by the outlet

- Type: Double

▶ **current** (readonly)

The current in amps.

- Type: Float

▶ **current_amps** (readonly) DEPRECATED **Use #current**

Amps drawn on the outlet

- Type: Double

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **max_active_power** (readonly)

Maximum Active Power reading at the outlet

- Type: Double

▶ **max_apparent_power** (readonly)

Maximum Apparent Power reading at the outlet

- Type: Double

▶ **max_current** (readonly)

The max_current in amps.

- Type: Float

▶ **max_current_amps** (readonly) DEPRECATED **Use #max_current**

Maximum current reading (Amps) on the outlet

- Type: Double

▶ **max_voltage** (readonly)

Maximum Voltage reading at the outlet

- Type: Double

▶ **max_watt_hour** (readonly) DEPRECATED

Maximum Watt Hour reading at the outlet

- Type: Double

▶ **min_active_power** (readonly)

Minimum Active Power reading at the outlet

- Type: Double

▶ **min_apparent_power** (readonly)

Minimum Apparent Power reading at the outlet

- Type: Double

▶ **min_current** (readonly)

The min_current field in amps

- Type: Float

▶ **min_current_amps** (readonly) DEPRECATED **Use #min_current**

Minimum current reading (Amps) on the outlet

- Type: Double

▶ **min_power_factor** (readonly)

Minimum power factor at the outlet.

- Type: Double

▶ **min_unutilized_capacity** (readonly)

The min_unutilized_capacity field

- Type: Float

▶ **min_voltage** (readonly)

Minimum Voltage reading at the outlet

- Type: Double

▶ **min_watt_hour** (readonly) DEPRECATED

Minimum Watt Hour reading at the outlet

- Type: Double

▶ **outlet_id** (readonly)

Outlet number on the PDU

- Type: Integer

▶ **pdu_id** (readonly)

Note: An EMX is treated the same as a PDU

The ID of the PDU that this resource is associated with

- Type: Integer

▶ **power_factor** (readonly)

Power factor at the outlet.

- Type: Double

▶ **reading_time** (readonly)

The date and time the reading was collected

- Type: String
- Sample Values: "2011/10/07 14:50:01 +0000"

▶ **unutilized_capacity** (readonly)

Unutilized capacity (Amps)

- Type: Float

▶ **volt_amp_hour** (readonly)

Total volt-amp-hours consumed by the outlet, if available.

- Type: Double

▶ **voltage** (readonly)

Voltage reading (Volts) at the outlet

- Type: Double

▶ **watt_hour** (readonly)

Watt-hours consumed by the outlet, if available

- Type: Double

OutletReadings*Rollup

OutletReadingsHourlyRollup

OutletReadingsDailyRollup

OutletReadingsMonthlyRollup

The OutletReadingsRollup resource summarizes the outlet readings power data over the rollup interval. The "*" may be replaced with the string Hourly, Daily, or Monthly. For example, OutletReadingsHourlyRollup.

Raw data is rolled up every hour, hourly roll-ups are in turn rolled up once a day, and daily roll-ups are rolled up once a month.

▶ **Examples:**

```
{
  "outlet_reading_hourly_rollup": {
    "average_active_power" : 0.0,
    "average_apparent_power" : 0.0,
    "average_current" : 0.0,
    "average_voltage" : 119.0,
    "id" : 1,
    "max_active_power" : 0.0,
    "max_apparent_power" : 0.0,
    "max_current" : 0.0,
    "max_voltage" : 119.0,
    "min_active_power" : 0.0,
    "min_apparent_power" : 0.0,
    "min_current" : 0.0,
    "min_voltage" : 119.0,
    "ordinal" : 44,
    "pdu_id" : 9,
    "reading_time" : "2011/10/20 18:00:00 -0400",
    "watt_hour" : 151007.0
  }
}
```

Attribute Details

▶ **average_active_power** (readonly)

Average Active Power (Watts) reading during rollup interval

- Type: Double

▶ **average_apparent_power** (readonly)

Average apparent power (VA) reading during rollup interval

- Type: Double

▶ **average_current** (readonly)

Average current (Amps) reading during rollup interval

- Type: Double

▶ **average_power_factor** (readonly)

Average power factor reading during rollup interval.

- Type: Double

▶ **average_unutilized_capacity** (readonly)

The average_unutilized_capacity field

- Type: Float

▶ **average_voltage** (readonly)

Average Voltage (V) reading during rollup interval

- Type: Double

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **max_active_power** (readonly)

Maximum Active Power (Watts) reading during rollup interval

- Type: Double

▶ **max_apparent_power** (readonly)

Maximum apparent power (VA) reading during rollup interval

- Type: Double

▶ **max_current** (readonly)

Maximum current (Amps) reading during rollup interval

- Type: Double

▶ **max_power_factor** (readonly)

Maximum power factor reading during rollup interval.

- Type: Double

▶ **max_unutilized_capacity** (readonly)

The max_unutilized_capacity field

- Type: Float

▶ **max_voltage** (readonly)

Maximum Voltage (V) reading during rollup interval

- Type: Double

▶ **min_active_power** (readonly)

Minimum Active Power (Watts) reading during rollup interval

- Type: Double

▶ **min_apparent_power** (readonly)

Minimum apparent power (VA) reading during rollup interval

- Type: Double

▶ **min_current** (readonly)

Lowest current (Amps) reading during rollup interval

- Type: Double

▶ **min_power_factor** (readonly)

Minimum power factor at the outlet.

- Type: Double

▶ **min_unutilized_capacity** (readonly)

The min_unutilized_capacity field

- Type: Float

▶ **min_voltage** (readonly)

Minimum Voltage (V) reading during rollup interval

- Type: Double

▶ **outlet_id** (readonly)

Outlet number on the PDU

- Type: Integer

► **pdu_id** (readonly)

Note: An EMX is treated the same as a PDU

The ID of the PDU that this resource is associated with

- Type: Integer

► **reading_time** (readonly)

The date and time the reading was collected

- Type: String
- Sample Values: "2011/10/07 14:50:01 +0000"

► **volt_amp_hour** (readonly)

Total volt-amp-hours consumed by the outlet, if available.

- Type: Double

► **rollup_interval** (readonly)

The frequency that raw readings are rolled up into this representation

- Type: Integer
- Sample Values: 1 = one hour, 2 = one day, 3 = one month

► **watt_hour** (readonly)

Watt Hour reading during rollup interval

- Type: Double

Pdu

The PDU resource describes a single PDU being managed by Power IQ.

► **Examples:**

```
{
  "pdu": {
    "caption": "36OutletPX2",
    "configured_inlet_voltage": null,
    "configured_outlet_voltage": null,
    "contact": "Mike Davidson",
    "custom_field_1": null,
    "custom_field_2": null,
    "default_connected_led_color": "3366ff",
    "default_disconnected_led_color": "ffff00",
    "description": "PX 020106",
    "dynamic_plugin_name": null,
    "external_key": "192.168.43.117",
    "firmware_version": "2.1.6.5-26030",
    "health": {
      "active_events_count": 0,
      "connectivity": "OK",
      "connectivity_explanation": "Most recent poll of the
target PDU was successful.",
      "events": "Good",
      "overall": "Good"
    },
    "manufacturer": "Raritan",
    "model": "PX2-5704U",
    "name": "36OutletPX2",
    "phase": "THREE_PHASE",
    "poller_plugin":
"com.raritan.polaris.plugins.pdu.raritan.px2.PduPolle
r",
    "proxy_index": null,
    "rated_amps": "24A",
    "rated_va": "15.0-17.3kVA",
    "rated_volts": "360-415V",
    "reading": { ... },
    "requires_manual_voltage": false,
    "serial_number": null,
    "snmp3_auth_level": null,
    "snmp3_enabled": false,
    "snmp3_user": null,
    "supports_bulk_configuration": false,
    "supports_data_logging": true,
    "supports_firmware_upgrades": true,
    "supports_outlet_power_control": true,
    "supports_outlet_renaming": true,
    "supports_readingsonly_poll": true,
```

```
    "supports_sensor_renaming": true,
    "supports_single_sign_on": true,
    "user_defined_phase": false
  }
}
```

Attribute Details

▶ **caption** (readonly)

PDU Name

- Type: String

▶ **contact** (readonly)

Contact name

- Type: String

▶ **default_connected_led_color**

Default LED color when in connected state in hexadecimal RGB.

- Type: String
- Sample Values: "FF0000"

▶ **default_disconnected_led_color**

Default LED color when in disconnected state in hexadecimal RGB.

- Type: String
- Sample Values: "FF0000"

▶ **description** (readonly)

MIB II SysDescr

- Type: String

▶ **dynamic_plugin_name** (readonly)

Name of the dynamic plugin, if any, being used by the PDU

- Type: String

▶ **external_key**

Note: External key is a unique string that identifies PDU's being managed by Power IQ. If left blank Power IQ assigns a default external key to each PDU

The external key that identifies the PDU.

- Type: String

▶ **firmware_version** (readonly)

PDU Firmware Version

- Type: String

▶ **health** (readonly)

PDU overall health status: Good, Warning, or Critical

- Type: String

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **inline_meter** (readonly)

Indicates if the PDU is an inline meter

- Type: Boolean

▶ **ip_address** (readonly)

IP Address

- Type: String

▶ **ipmi_username**

The username of the user used to log into the Web UI of the PDU if it is a Raritan PDU

- Type: String
- Invisible and writable

▶ **ipmi_password**

The password of the user used to log into the Web UI of the PDU if it is a Raritan PDU

- Type: String
- Invisible and writable

▶ **location** (readonly)

MIB II SysLocation

- Type: Location

▶ **manufacturer** (readonly)

PDU Manufacturer

- Type: String

▶ **model** (readonly)

PDU Model

- Type: String

▶ **name** (readonly)

Name of the PDU

- Type: String

▶ **phase** (readonly)

Number of phases in the PDU: SINGLE_PHASE or THREE_PHASE

- Type: String

▶ **proxy_index** (readonly)

Subtending unit ID applicable only to daisy-chained and console server connected PDU units

- Type: Integer

▶ **rated_amps** (readonly)

Rated Amps (A) on the PDU

- Type: Double

▶ **rated_va** (readonly)

Rated Volts (V) on the PDU

- Type: Double

▶ **rated_volts** (readonly)

The Rated Volts (V) on the PDU

- Type: Double

▶ **reading** (readonly)

The latest reading associated with the outlet.

- Type: Object

105

▶ **requires_manual_voltage** (readonly)

Indicates whether a PDU requires manual voltage

- Type: Boolean

▶ **serial_number** (readonly)

PDU Serial Number

- Type: String

▶ **snmp3_auth_level**

The snmp3_auth_level field

- Type: Integer

▶ **snmp3_enabled**

Indicates SNMP version 3 is enabled on the PDU.

- Type: Boolean

▶ **supports_bulk_configuration** (readonly)

Indicates whether the PDU supports bulk configuration

- Type: Boolean

▶ **supports_data_logging** (readonly)

Indicates whether the PDU supports data logging

- Type: Boolean

▶ **supports_firmware_upgrades** (readonly)

Indicates whether the PDU supports firmware upgrades

- Type: Boolean

▶ **supports_outlet_power_control** (readonly)

Indicates if a PDU supports outlet power control

- Type: Boolean

▶ **supports_outlet_renaming** (readonly)

Indicates if the PDU allows its outlets to be renamed

- Type: Boolean

▶ **supports_readingsonly_poll** (readonly)

Indicates whether the PDU supports resadings only poll

- Type: Boolean

▶ **supports_sensor_renaming** (readonly)

Indicates whether the PDU allows sensors to be renamed

- Type: Boolean

▶ **supports_single_sign_on** (readonly)

Indicates whether the PDU supports single sign on

- Type: Boolean

▶ **user_defined_phase** (readonly)

User defined phase value

- Type: Integer

Rack

Physical Rack in a data center. Part of the data center hierarchy that can be used to model your site.

▶ **Examples:**
```
{"rack" : {
 "id" : 2,
 "name" :"Rack 2",
 "external_key" :"Rack -- 2",
 "capacity" :1.0
 }}
```
Attribute Details

▶ **capacity**

User-defined power capacity in kW

- Type: Double
- Sample Values:1.0

▶ **external_key**

A code used to uniquely identify this resource

- Type: String

> ▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

> ▶ **name**

The name of the resource

- Type: String

> ▶ **space_id**

The space_id field.

- Type: String

RackUnit

RackUnit attached to an AssetStrip, that is being managed by Power IQ.

> ▶ **Examples:**

```
{"rack_unit": {
 "id":1,
 "asset_strip_id":1,
 "tag_id":"000013DBDA6F",
 "ordinal":1,
 "led_state":"on",
 "led_mode":"automatic",
 "led_color":"ff0000",
 "created_at":"2011/10/07 14:50:01 +0000",
 "updated_at":"2011/10/07 14:50:01 +0000"
 }
}
```

Attribute Details

> ▶ **asset_strip_id** (readonly)

The AssetStrip this RackUnit is associated with

- Type: Integer

> ▶ **blade_extension_size** (readonly)

The size of the blade extension if it exists.

- Type: Integer

▶ **created_at** (readonly)

The date and time this resource was created

- Type: String

- Sample Values: "2011/10/07 14:50:00 +0000"

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **led_color**

The color of a RackUnit LED connected to the AssetStrip, in hexadecimal RGB.

- Type: String

- Sample Values: "FF0000"

▶ **led_mode**

The mode the LED is operating in

- Type: String

- Sample Values: automatic, manual

▶ **led_state**

The state the LED is operating in

- Type: String

- Sample Values: on, off, blinking, slow_blinking

▶ **name**

The name of the resource.

- Type: String

▶ **ordinal** (readonly)

The position that this RackUnit resource occupies in the AssetStrip

- Type: Integer

▶ **rack_unit_position (readonly)**

The position that this RackUnit resource claims to occupy in the AssetStrip.

- Type: Integer

► **rack_unit_type**

The type of rack unit.

- Type: String
- Sample Values: single, blade

► **tag_family**

The tag family string of the rack unit.

- Type: String

► **tag_id** (readonly)

Unique id of the attached asset tag that is plugged into this asset strip. The tag_id is built directly into the hardware of the asset tag.

- Type: String

► **updated_at** (readonly)

Note: The timezone for all date and time fields is always UTC (+0000). Time granularity is returned in seconds but internally stored to the usec. Be aware of this when performing searches against particular times.

- Type: String
- Sample Values: "2011/10/07 14:50:01 +0000"

Room

Physical Room in a data center. Part of the data center hierarchy that can be used to model your site.

Examples:
```
{"room" : {
 "id" : 2,
 "name" :"Room 2",
 "external_key" :"Room -- 2",
 "capacity" :1.0
}}
```
Attribute Details

► **capacity**

User-defined power capacity in kW

- Type: Double
- Sample Values: .0

> ▶ **external_key**

A code used to uniquely identify this resource

- ▪ Type: String

> ▶ **id** (readonly)

An automatically generated ID for this resource

- ▪ Type: Integer

> ▶ **name**

The name of the resource

- ▪ Type: String

Row

Physical Row in a data center. Part of the data center hierarchy that can be used to model your site.

> ▶ **Examples:**

```
{"row" : {
 "id" : 2,
 "name" :"Row 2",
 "external_key" :"Row -- 2",
 "capacity" :1.0
}}
```

Attribute Details

> ▶ **capacity**

User-defined power capacity in kW

- ▪ Type: Double
- ▪ Sample Values: .0

> ▶ **external_key**

A code used to uniquely identify this resource

- ▪ Type: String

> ▶ **id** (readonly)

An automatically generated ID for this resource

- ▪ Type: Integer

111

▶ **name**

The name of the resource

- Type: String

Sensor

A Sensor is attached to a PDU being managed by Power IQ.

▶ **Examples:**

```
{"sensor":
  {
     "id":10,
    "pdu_id":28,
    "attribute_name":"HUMIDITY",
    "label":"H6_45.246",
    "removed":null,
    "position":"INLET",
    "reading":{},
    "state":{}
  }
}
```

Attribute Details

▶ **attribute_name** (readonly)

The type of sensor

- Type: String

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **label** (readonly)

Sensor's label as gathered from the PDU

- Type: String

▶ **ordinal** (readonly)

Position of the sensor on the PDU, as gathered from the PDU.

- Type: Integer

▶ **pdu_id** (readonly)

Note: An EMX is treated the same as a PDU

The ID of the PDU that this resource is associated with

- Type: Integer

▶ **pdu_sensor_id** (readonly) DEPRECATED

Physical ID of the sensor on the PDU, as gathered from the pdu

- Type: Integer

▶ **position** (readonly)

Sensor's position

- Type: String
- Sample Values: INLET, OUTLET, EXTERNAL

▶ **reading** (readonly)

The reading field

- Type: Hash

▶ **removed** (readonly)

When the sensor was removed from the system (always null for current sensors)

- Type: String

▶ **state** (readonly)

The reading field

- Type: Hash

SensorReadings

Note: SensorReadings are periodically purged from the system

The SensorReadings are created from the data collected from PDU sensors. A data record is added for each sensor polled. This data is summarized in the SensorReadingsRollup resource.

▶ **Examples:**

```
{"sensor_reading":
 {
    "id":4950,
    "reading_time":"2011/10/19 14:39:34 +0000",
    "value":44.0,
    "sensor_id":3,
    "max_value":null,
    "min_value":null,
    "uom":"%"
 }
}
```

Attribute Details

▶ **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

▶ **max_value** (readonly)

The max_value field

- Type: Float

▶ **min_value** (readonly)

The min_value field

- Type: Float

▶ **reading_time** (readonly)

The date and time the reading was collected

- Type: String
- Sample Values: "2011/10/07 14:50:01 +0000"

▶ **sensor_id** (readonly)

The ID of the sensor that this resource is associated with

- Type: Integer

114

► **uom** (readonly)

The unit of measure used by the system when calculating reading values return

- Type: String

► **value** (readonly)

The value field

- Type: String

SensorReadings*Rollups

SensorReadings*Rollup summarize the sensor readings power data over the rollup interval. The "*" may be replaced with the string Hourly, Daily, or Monthly. For example, SensorReadingsHourlyRollup.

Raw data is rolled up every hour, hourly roll-ups are in turn rolled up once a day, and daily roll-ups are rolled up once a month.

► **Examples:**

```
{
  "sensor_reading_hourly_rollup": {
    "id": 1,
    "rollup_interval": 1,
    "reading_time": "2011/10/07 14:00:00 +0000",
    "min_value": 31.0,
    "max_value": 31.0,
    "average_value": 31.0,
    "sensor_id": 45
  }
}
```

Attribute Details

► **average_value** (readonly)

The average_value field

- Type: Float

► **id** (readonly)

An automatically generated ID for this resource

- Type: Integer

► **max_value** (readonly)

The max_value field

- Type: Float

▶ **min_value** (readonly)

The min_value field

- Type: Float

▶ **reading_time** (readonly)

The date and time the reading was collected

- Type: String
- Sample Values: "2011/10/07 14:50:01 +0000"

▶ **sensor_id** (readonly)

The ID of the sensor that this resource is associated with

- Type: Integer

SystemInfo

SystemInfo displays the settings for Power IQ (which can only be set in the UI), as well as version and licensing information.

▶ **Examples:**

```
{"system_info":{
 "poweriq_version":"3.1.0-15",
 "uuid":"f24a8aa6-4c0f-4afd-bbaf-e527bea737c0",
 "current_time":"2011/10/19 15:39:45 +0000",
 "svn_branch":"trunk",
 "configuration":{
    "login_header":"Unauthorized Access Warning",
    "login_body":"Access to this computer is prohibited
unless authorized.<br/>Accessing programs or data
unrelated to your job is prohibited.",
    "session_timeout":30,
    "px_minimum_version":"1.1.0-6684",
    "site_locale":"en-US",
    "eula_accepted":true,
    "sso_enabled":false,
    "ntp_enabled":false,
    "time_zone":"UTC",
    "time_zone_offset":0,
    "temperature_unit":"C",
    "currency":"$",
    "data_center_custom_field_1":"Custom Field 1",
    "data_center_custom_field_2":"Custom Field 2",
    "device_custom_field_1":"Custom Field 1",
    "device_custom_field_2":"Custom Field 2",
    "snmp_version":"1/2c",
    "enable_power_control":false,
    "require_power_control_audit_message":false,
    "enable_browser_power_control":false,
    "enable_scheduled_power_control":false,
    "enable_web_api":false,
    "enable_event_notifications":false,
    "smtp_server":null, "smtp_port":null,
    "smtp_auth_type":null,
    "smtp_username":null,
    "from_email":null,
    "smtp_encryption_method":"NONE",
    "pdu_label":"IP_ADDRESS",
    "password_min_length":8,
    "password_max_length":40,
    "password_requires_one_lowercase":true,
    "password_requires_one_uppercase":true,
    "password_requires_one_numeric":true,
    "password_requires_one_special":true,
    "enable_web_api_user":false,
    "enable_custom_logo":false,
```

117

```
                              "allow_html_portlets":false,
                              "browser_session_polling_interval":300,
                              "remote_storage_enabled":false,
                              "remote_storage_protocol":null,
                              "remote_storage_host":null,
                              "remote_storage_port":null,
                              "remote_storage_ftp_user":null,
                              "remote_storage_directory":null,
                              "remote_storage_s3_ssl_only":true,
                              "remote_storage_push_csv":false,
                              "remote_storage_push_backup":true,
                              "pdu_custom_field_1":"Custom Field 1",
                              "pdu_custom_field_2":"Custom Field 2"
                           },
                           "licensing":{
                              "device_limit":500,
                              "customer_name":"Internal Use Only",
                              "enable_events":false
                           }
                        }}
```

Attribute Details

▶ **configuration** (readonly)

Configuration information for Power IQ. The configuration is can only be set via the Power IQ UI

- ▪ Type: Configuration**)**

▶ **current_time** (readonly)

The current time configured on Power IQ

- ▪ Type: String
- ▪ Sample Values: "2011/10/07 14:50:00 +0000"

▶ **licensing** (readonly)

Licensing information

- ▪ Type: Licensing

▶ **poweriq_version** (readonly)

The version of Power IQ that is running

- ▪ Type: String

► **uuid** (readonly)

A UUID assigned to this instance of Power IQ

- Type: String

Addendum B: Route Reference

The following table is a synopsis of the available URLs and methods available for all the resources in Power IQ in alphabetical order.

Method	URL	Deprecated?
GET	/api/v2/aisles	
POST	/api/v2/aisles	
GET	/api/v2/aisles/:id/children	
GET	/api/v2/aisles/:id/descendants	
GET	/api/v2/aisles/:id	
PUT	/api/v2/aisles/:id	
DELETE	/api/v2/aisles/:id	
PUT	/api/v2/aisles/:id/move_to	
POST	/api/v2/aisles/:id/power_control	
GET	/api/v2/aisles/:id/siblings	
GET	/api/v2/asset_strips/:asset_strip_id/rack_units	
GET	/api/v2/asset_strips	
GET	/api/v2/asset_strips/:id	
PUT	/api/v2/asset_strips/:id	
GET	/api/v2/circuit_breaker_readings	
GET	**/api/v2/circuit_breaker_readings_rollups/**	**DEPRECATED**
GET	/api/v2/circuit_breaker_readings_rollups/hourly	
GET	/api/v2/circuit_breaker_readings_rollups/daily	
GET	/api/v2/circuit_breaker_readings_rollups/monthly	
GET	/api/v2/data_centers	
POST	/api/v2/data_centers	
GET	/api/v2/data_centers/:id/children	

Method	URL	Deprecated?
GET	/api/v2/data_centers/:id/descendants	
GET	/api/v2/data_centers/:id	
PUT	/api/v2/data_centers/:id	
DELETE	/api/v2/data_centers/:id	
PUT	/api/v2/data_centers/:id/move_to	
GET	/api/v2/data_centers/:id/siblings	
GET	/api/v2/devices/:device_id/outlets	
GET	/api/v2/devices	
POST	/api/v2/devices	
GET	/api/v2/devices/:id	
PUT	/api/v2/devices/:id	
DELETE	/api/v2/devices/:id	
PUT	/api/v2/devices/:id/move_to	
POST	/api/v2/devices/:id/power_control	
POST	/api/v2/device_groups/power_control	
PUT	/api/v2/events/clear_batch	
GET	/api/v2/events	
PUT	/api/v2/events/:id/clear	
GET	/api/v2/events/:id	
GET	/api/v2/floors	
POST	/api/v2/floors	
GET	/api/v2/floors/:id/children	
GET	/api/v2/floors/:id/descendants	
GET	/api/v2/floors/:id	
PUT	/api/v2/floors/:id	
DELETE	/api/v2/floors/:id	
PUT	/api/v2/floors/:id/move_to	
GET	/api/v2/floors/:id/siblings	

Method	URL	Deprecated?
GET	/api/v2/inlets	
GET	/api/v2/inlets/:id/inlet_poles	
GET	/api/v2/inlets/:id/readings	
GET	/api/v2/inlets/:id/readings_rollups/hourly	
GET	/api/v2/inlets/:id/readings_rollups/daily	
GET	/api/v2/inlets/:id/readings_rollups/monthly	
GET	/api/v2/inlet_readings	
GET	/api/v2/inlet_readings_rollups/hourly	
GET	/api/v2/inlet_readings_rollups/daily	
GET	/api/v2/inlet_readings_rollups/monthly	
GET	/api/v2/inlet_pole_readings	
GET	/api/v2/inlet_pole_readings_rollups/hourly	
GET	/api/v2/inlet_pole_readings_rollups/daily	
GET	/api/v2/inlet_pole_readings_rollups/monthly	
GET	/api/v2/inlet_poles	
GET	/api/v2/inlet_poles/:id/readings	
GET	/api/v2/inlet_poles/:id/readings_rollups/hourly	
GET	/api/v2/inlet_poles/:id/readings_rollups/daily	
GET	/api/v2/inlet_poles/:id/readings_rollups/monthly	
GET	/api/v2/job_messages	
GET	/api/v2/job_messages/:id	
GET	/api/v2/jobs/:id	
GET	/api/v2/jobs/:job_id/messages	
GET	/api/v2/outlet_readings	
GET	**/api/v2/outlet_readings_rollups**	**DEPRECATED**
GET	/api/v2/outlet_readings_rollups/hourly	

Method	URL	Deprecated?
GET	/api/v2/outlet_readings_rollups/daily	
GET	/api/v2/outlet_readings_rollups/monthly	
GET	/api/v2/outlets	
GET	/api/v2/outlets/:id	
PUT	/api/v2/outlets/:id	
GET	/api/v2/outlets/:outlet_id/readings	
GET	**/api/v2/outlets/:outlet_id/readings_rollups**	**DEPRECATED**
GET	/api/v2/outlets/:outlet_id/readings_rollups/hourly	
GET	/api/v2/outlets/:outlet_id/readings_rollups/daily	
GET	/api/v2/outlets/:outlet_id/readings_rollups/monthly	
GET	/api/v2/outlets/:outlet_id/events	
POST	/api/v2/outlets/power_control	
PUT	/api/v2/outlets/rename_batch	
POST	/api/v2/pdus/create_batch	
DELETE	/api/v2/pdus/destroy_batch	
GET	/api/v2/pdus	
POST	/api/v2/pdus	
GET	/api/v2/pdus/:id	
PUT	/api/v2/pdus/:id	
DELETE	/api/v2/pdus/:id	
GET	/api/v2/pdus/:id/inlets	
PUT	/api/v2/pdus/:id/move_to	
PUT	/api/v2/pdus/:id/rescan	
GET	/api/v2/pdus/:pdu_id/asset_strips	
GET	/api/v2/pdus/:pdu_id/events	
GET	/api/v2/pdus/:pdu_id/outlets	
GET	/api/v2/pdus/:pdu_id/sensors	

Method	URL	Deprecated?
GET	/api/v2/racks	
POST	/api/v2/racks	
GET	/api/v2/racks/:id/children	
GET	/api/v2/racks/:id/descendants	
GET	/api/v2/racks/:id	
PUT	/api/v2/racks/:id	
DELETE	/api/v2/racks/:id	
PUT	/api/v2/racks/:id/move_to	
POST	/api/v2/racks/:id/power_control	
GET	/api/v2/racks/:id/siblings	
GET	/api/v2/rack_units	
GET	/api/v2/rack_units/:id	
PUT	/api/v2/rack_units/:id	
GET	/api/v2/rooms	
POST	/api/v2/rooms	
GET	/api/v2/rooms/:id/children	
GET	/api/v2/rooms/:id/descendants	
GET	/api/v2/rooms/:id	
PUT	/api/v2/rooms/:id	
DELETE	/api/v2/rooms/:id	
PUT	/api/v2/rooms/:id/move_to	
POST	/api/v2/rooms/:id/power_control	
GET	/api/v2/rooms/:id/siblings	
GET	/api/v2/rows	
POST	/api/v2/rows	
GET	/api/v2/rows/:id/children	
GET	/api/v2/rows/:id/descendants	
GET	/api/v2/rows/:id	
PUT	/api/v2/rows/:id	

Method	URL	Deprecated?
DELETE	/api/v2/rows/:id	
PUT	/api/v2/rows/:id/move_to	
POST	/api/v2/rows/:id/power_control	
GET	/api/v2/rows/:id/siblings	
GET	/api/v2/sensor_readings	
GET	/api/v2/sensor_readings_rollups	
GET	/api/v2/sensor_readings_rollups/hourly	
GET	/api/v2/sensor_readings_rollups/daily	
GET	/api/v2/sensor_readings_rollups/monthly	
GET	/api/v2/sensors	
GET	/api/v2/sensors/:id	
GET	/api/v2/sensors/:sensor_id/events	
GET	/api/v2/sensors/:sensor_id/readings	
GET	**/api/v2/sensors/:sensor_id/readings_rollups**	**DEPRECATED**
GET	/api/v2/sensors/:sensor_id/readings_rollups/hourly	
GET	/api/v2/sensors/:sensor_id/readings_rollups/daily	
GET	/api/v2/sensors/:sensor_id/readings_rollups/monthly	
GET	/api/v2/system_info	

Addendum C: Searching Reference

Many of the resources provided in the Power IQ REST API provide searching capabilities. For an understanding of which resources have searching capabilities, see the API section of this document. Most resources allow for searching.

Searching is performed by passing in modified parameters to the request. Most resources allow for searching, but searching can only be applied to the listing/index route. For example, to perform a search on the event resource, this route must be used:

```
GET /api/v2/events
```

Searching works by creating URL parameter strings that are a combination of the resource field name with various modifiers appended to it. Combining multiple parameters together is effectively performing an AND on the search.

► Example 1

Retrieve all events for the PDU with an ID of 50:

```
GET /api/v2/events?pdu_id_eq=50
```

In the example above:

pdu_id is a valid field in the events resource

_eq is a valid search modifier

50 is the input to the query, the ID of the PDU

► Example 2

Retrieve all events for the PDU with an ID of 50, that occurred on or after 2011/10/19 14:00:00:

```
GET
/api/v2/events?pdu_id_eq=50&created_at_gt=2011/10/19%2014:00:00
```

In the example above:

created_at is a valid field in the events resource

_gt is a valid search modifier

2011/10/19%2014:00:00 is the input to the query (url encoded)

► Example 3

Retrieve all events with an event_config_id of 72 or 62:

```
GET
/api/v2/events?event_config_id_in[]=72&event_config_id_in[]=62
```

In the example above:

event_config_id is a valid field in the events resource

_in is a valid search modifier

[] create an array so that multiple inputs can be used for the same parameter

62,72 is the input to the query

Modifiers

The section below lists all the modifiers supported for generating url queries. Note that the modifiers depend on the field type.

▶ **All data types**

equals (alias: eq) - Just as it sounds.

does_not_equal (aliases: ne, noteq) - The opposite of equals, oddly enough.

in - Takes an array, matches on equality with any of the items in the array.

not_in (aliases: ni, notin) - Like above, but negated.

is_null - The column has an SQL NULL value.

is_not_null - The column contains anything but NULL.

▶ **Strings**

contains (aliases: like, matches) - Substring match.

does_not_contain (aliases: nlike, nmatches) - Negative substring match.

starts_with (alias: sw) - Match strings beginning with the entered term.

does_not_start_with (alias: dnsw) - The opposite of above.

ends_with (alias: ew) - Match strings ending with the entered term.

does_not_end_with (alias: dnew) - Negative of above.

▶ **Numbers, dates, and times**

greater_than (alias: gt) - Greater than.

greater_than_or_equal_to (aliases: gte, gteq) - Greater than or equal to.

less_than (alias: lt) - Less than.

less_than_or_equal_to (aliases: lte, lteq) - Less than or equal to.

▶ **Booleans**

is_true - Is true

is_false - The complement of is_true.

▶ **Non-boolean**

is_present - Not NULL or the empty string.

is_blank - Returns values where the fields is NULL or the empty string.

Addendum D: Asynchronous Modes

Some actions on resources, by their nature, take an extended period of time to complete. In cases where an action requires a long time to complete, that action can optionally (or in some cases required to) be executed asynchronously.

Job resources are used to handle long running requests that either must be, or can optionally be, executed asynchronously. An action on any resource will return a job resource instead of the usuall representation of itself, if it supports the async parameter.

For an understanding of which resources and actions support or require asynchronous execution, see the API section of this document.

Some resource actions will always be run asynchronously, and will be noted as such.

Addendum E: Response Codes

CODE	INTERPRETATION
200 (ok)	Everything worked as expected
400 (bad request)	Usually missing or invalid parameters or body content in request, or failure to set Content-Type and Accept headers to application/json.
401 (unauthorized)	No valid authentication provide, or authentication provided does not grant access to the requested resource
404 (not found)	The requested resource and or action does not exist
406 (unacceptable)	Usually indicates a request that does not set the Content-Type to application/json, or a request for a resource in a format that is not supported, such as xml or html.
422 (unprocessable entity)	Usually indicates that an asynchronous request failed
500 (internal server error)	Something went wrong on the Power IQ server

Index

▶ U.S./Canada/Latin America

Monday - Friday
8 a.m. - 6 p.m. ET
Phone: 800-724-8090 or 732-764-8886
For CommandCenter NOC: Press 6, then Press 1
For CommandCenter Secure Gateway: Press 6, then Press 2
Fax: 732-764-8887
Email for CommandCenter NOC: tech-ccnoc@raritan.com
Email for all other products: tech@raritan.com

▶ China

Beijing
Monday - Friday
9 a.m. - 6 p.m. local time
Phone: +86-10-88091890

Shanghai
Monday - Friday
9 a.m. - 6 p.m. local time
Phone: +86-21-5425-2499

GuangZhou
Monday - Friday
9 a.m. - 6 p.m. local time
Phone: +86-20-8755-5561

▶ India

Monday - Friday
9 a.m. - 6 p.m. local time
Phone: +91-124-410-7881

▶ Japan

Monday - Friday
9:30 a.m. - 5:30 p.m. local time
Phone: +81-3-3523-5991
Email: support.japan@raritan.com

▶ Europe

Europe
Monday - Friday
8:30 a.m. - 5 p.m. GMT+1 CET
Phone: +31-10-2844040
Email: tech.europe@raritan.com

United Kingdom
Monday - Friday
8:30 a.m. to 5 p.m. GMT
Phone +44(0)20-7090-1390

France
Monday - Friday
8:30 a.m. - 5 p.m. GMT+1 CET
Phone: +33-1-47-56-20-39

Germany
Monday - Friday
8:30 a.m. - 5:30 p.m. GMT+1 CET
Phone: +49-20-17-47-98-0
Email: rg-support@raritan.com

▶ Melbourne, Australia

Monday - Friday
9:00 a.m. - 6 p.m. local time
Phone: +61-3-9866-6887

▶ Taiwan

Monday - Friday
9 a.m. - 6 p.m. GMT -5 Standard -4 Daylight
Phone: +886-2-8919-1333
Email: support.apac@raritan.com