USER GUIDE

Ruckus SmartCell Insight API User Guide, 5.4

Supporting SmartCell Insight 5.4

Part Number: 800-72532-001 Rev A
Publication Date: February 2020

Copyright, Trademark and Proprietary Rights Information

© 2020 CommScope, Inc. All rights reserved.

No part of this content may be reproduced in any form or by any means or used to make any derivative work (such as translation, transformation, or adaptation) without written permission from CommScope, Inc. and/or its affiliates ("CommScope"). CommScope reserves the right to revise or change this content from time to time without obligation on the part of CommScope to provide notification of such revision or change.

Export Restrictions

These products and associated technical data (in print or electronic form) may be subject to export control laws of the United States of America. It is your responsibility to determine the applicable regulations and to comply with them. The following notice is applicable for all products or technology subject to export control:

These items are controlled by the U.S. Government and authorized for export only to the country of ultimate destination for use by the ultimate consignee or end-user(s) herein identified. They may not be resold, transferred, or otherwise disposed of, to any other country or to any person other than the authorized ultimate consignee or end-user(s), either in their original form or after being incorporated into other items, without first obtaining approval from the U.S. government or as otherwise authorized by U.S. law and regulations.

Disclaimer

THIS CONTENT AND ASSOCIATED PRODUCTS OR SERVICES ("MATERIALS"), ARE PROVIDED "AS IS" AND WITHOUT WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED. TO THE FULLEST EXTENT PERMISSIBLE PURSUANT TO APPLICABLE LAW, COMMSCOPE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, TITLE, NON-INFRINGEMENT, FREEDOM FROM COMPUTER VIRUS, AND WARRANTIES ARISING FROM COURSE OF DEALING OR COURSE OF PERFORMANCE. CommScope does not represent or warrant that the functions described or contained in the Materials will be uninterrupted or error-free, that defects will be corrected, or are free of viruses or other harmful components. CommScope does not make any warranties or representations regarding the use of the Materials in terms of their completeness, correctness, accuracy, adequacy, usefulness, timeliness, reliability or otherwise. As a condition of your use of the Materials, you warrant to CommScope that you will not make use thereof for any purpose that is unlawful or prohibited by their associated terms of use.

Limitation of Liability

IN NO EVENT SHALL COMMSCOPE, COMMSCOPE AFFILIATES, OR THEIR OFFICERS, DIRECTORS, EMPLOYEES, AGENTS, SUPPLIERS, LICENSORS AND THIRD PARTY PARTNERS, BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER, EVEN IF COMMSCOPE HAS BEEN PREVIOUSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, WHETHER IN AN ACTION UNDER CONTRACT, TORT, OR ANY OTHER THEORY ARISING FROM YOUR ACCESS TO, OR USE OF, THE MATERIALS. Because some jurisdictions do not allow limitations on how long an implied warranty lasts, or the exclusion or limitation of liability for consequential or incidental damages, some of the above limitations may not apply to you.

Trademarks

ARRIS, the ARRIS logo, CommScope, Ruckus, Ruckus Wireless, Ruckus Networks, Ruckus logo, the Big Dog design, BeamFlex, ChannelFly, EdgeIron, FastIron, HyperEdge, ICX, IronPoint, OPENG, SmartCell, Unleashed, Xclaim, and ZoneFlex are trademarks of CommScope, Inc. and/or its affiliates. Wi-Fi Alliance, Wi-Fi, the Wi-Fi logo, Wi-Fi Certified, the Wi-Fi CERTIFIED logo, Wi-Fi Protected Access, the Wi-Fi Protected Setup logo, Wi-Fi Protected Setup, Wi-Fi Multimedia and WPA2 and WMM are trademarks or registered trademarks of Wi-Fi Alliance. All other trademarks are the property of their respective owners.

Contents

About This Guide

This *SmartCell Insight API User Guide* provides instructions about how the Ruckus Wireless™ SmartCell Insight (SCI) APIs work to access the various functionalities provided by the core SCI engine.

This guide is written for service operators and system administrators who are responsible for managing, configuring, and troubleshooting Wi-Fi networks. It assumes basic working knowledge of local area networks, wireless networking, and wireless devices.

NOTE
Refer to the release notes shipped with your product to be aware of certain challenges when upgrading to this release.

Most user guides and release notes are available in Adobe Acrobat Reader Portable Document Format (PDF) or HTML on the Ruckus Wireless Support Web site at https://support.ruckuswireless.com/contact-us.

Document Conventions

Document Conventions and Document Conventions list the text and notice conventions that are used throughout this guide.

TABLE 1 Text conventions

Convention	Description	Example
`message phrase`	Represents messages displayed in response to a command or a status	`[Device Name]` >
user input	Represents information that you enter	`[Device Name]` > **set ipaddr 10.0.0.12**
user interface controls	Keyboard keys, software buttons, and field names	Click **Create New**
Start > **All Programs**	Represents a series of commands, or menus and submenus	Select **Start** > **All Programs**
ctrl+V	Represents keyboard keys pressed in combination	Press **ctrl+V** to paste the text from the clipboard.
screen or page names		Click **Advanced Settings**. The **Advanced Settings** page appears.
command name	Represents CLI commands	
parameter name	Represents a parameter in a CLI command or UI feature	
variable name	Represents variable data	*{ZoneDirectorID}*
`filepath`	Represents file names or URI strings	`http://ruckuswireless.com`

TABLE 2 Notice conventions

Notice type	Description
NOTE	Information that describes important features or instructions
CAUTION	Information that alerts you to potential loss of data or potential damage to an application, system, or device

TABLE 2 Notice conventions (continued)

Notice type	Description
WARNING	Information that alerts you to potential personal injury

Related Documentation

For a complete list of documents that accompany this release, refer to the Release Notes.

Documentation Feedback

Ruckus Wireless is interested in improving its documentation and welcomes your comments and suggestions.

You can email your comments to Ruckus Wireless at: docs@ruckuswireless.com

When contacting us, please include the following information:

- Document title
- Document part number (on the cover page)
- Page number (if appropriate)

Overview

SmartCell Insight (SCI) is a Big Data analytics and reporting engine that provides deep visibility into the performance and operational statistics of your Ruckus Wireless WiFi infrastructure.

SmartCell Insight (SCI) is designed to collect data from Ruckus network equipment, analyze that data, and then present it using a wide variety of standard and custom reports. SCI provides visibility, analytics and reports about network transmission statistics, equipment status and user traffic. It also provides details about the devices and applications that are used on the network, so that decision-makers can make better informed decisions about what types of devices and content their customers are using and will be using more of in the future.

SCI provides a rich set of APIs to access the various functionality provided by the core SCI engine.

Ths manual includes the following information:

- Report Types on page 7
- Common Tasks on page 8
- Using the API Dialog Box on page 9
- Using SCI API Explorer on page 11:
  - Generating the Access Token on page 13
  - Querying to Obtain Report IDs on page 15
  - Querying to Obtain Section IDs of a Specific Report on page 18
  - Querying the Data Endpoint on page 20
  - Logging in to API Explorer Programmatically on page 23

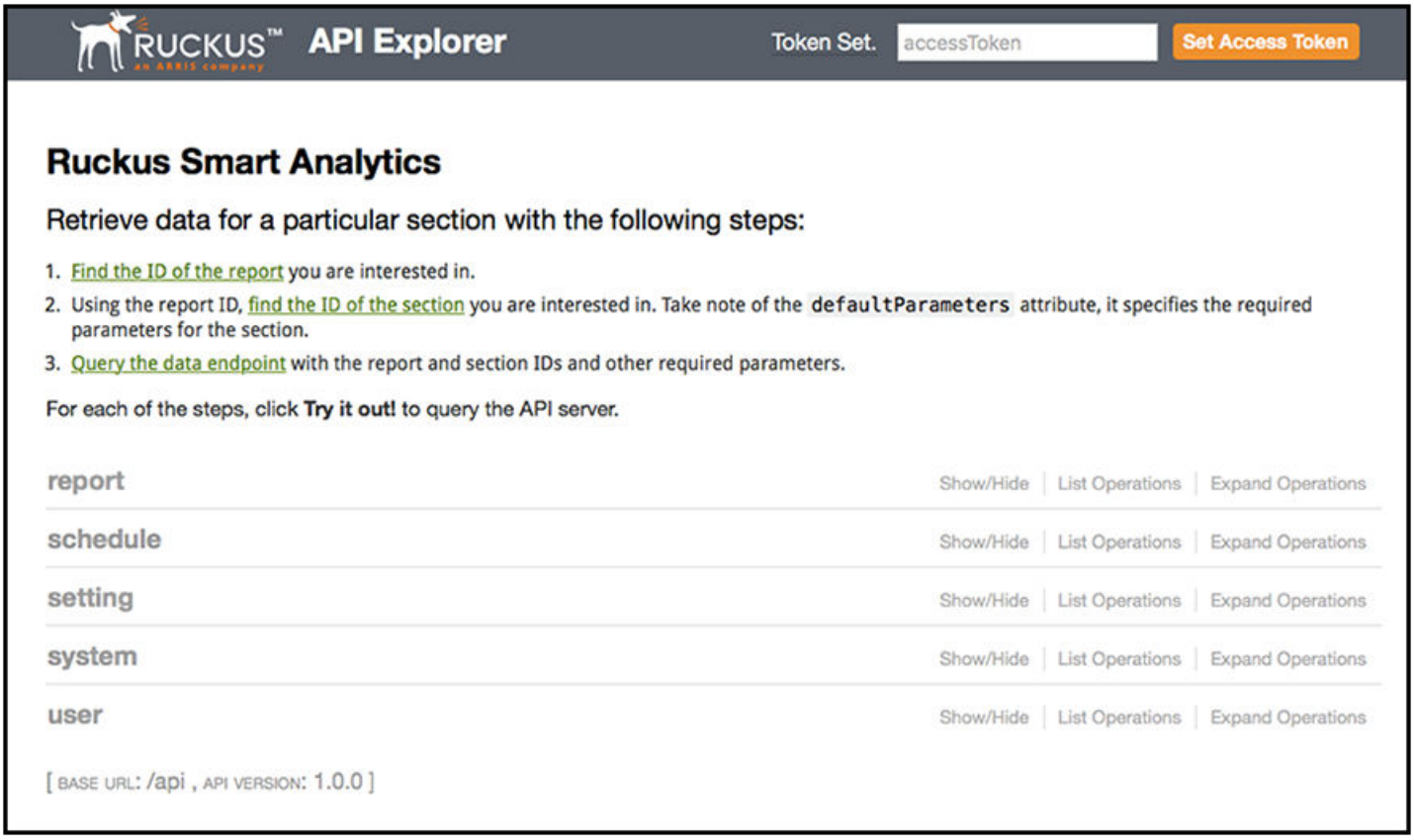
NOTE
Refer to the "Ruckus Smart Analytics" section of this manual for information about how to use each of the APIs.

Report Types

This section lists and describes the report types available in SCI.

When you first enter the SCI API Explorer by entering `https://<your SCI IP address>/explorer` in your web browser. the following screen is displayed:

FIGURE 1 Ruckus Smart Analytics Screen when Entering API Explorer



Report types available in SCI are described in the following table:

Report Type	Description
Report	Report type is for various reports available in the application such as Network, WLAN, and Clients.
Schedule	This is to schedule automatic report generation and delivery. You can also specify occurrences for a particular schedule. (Refer to the SCI user guide for details).
Setting	This report type is for system level settings including SMTP settings.
System	This report type is based on all the controllers that report to the SCI as data sources.
User	The user usage report returns activities pertaining to SCI across the user's accounts.

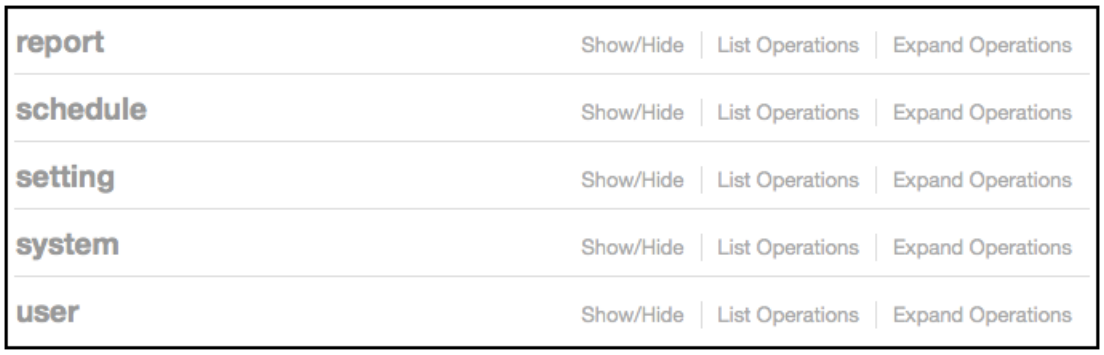
Common Tasks

SCI's API allows you to build your custom specified reports, based on the available parameters. Each of these parameters has the following common tasks.

- Show / Hide: This toggle command shows or hides the rows dynamically in a table.
- List Operations: This command displays the list of HTTP verbs such as GET, PUT, POST, HEAD or DELETE row dynamically in a table.

- Expand Operations: This command expands the API dialog box for each listed operation. Use List Operations to contract the view.

FIGURE 2 Common tasks



Using the API Dialog Box

Use the API dialog box to view and modify the messages to generate your API reports.

On clicking each of the parameters and the resource URL the API dialog box is displayed.

FIGURE 3 API Dialog Box

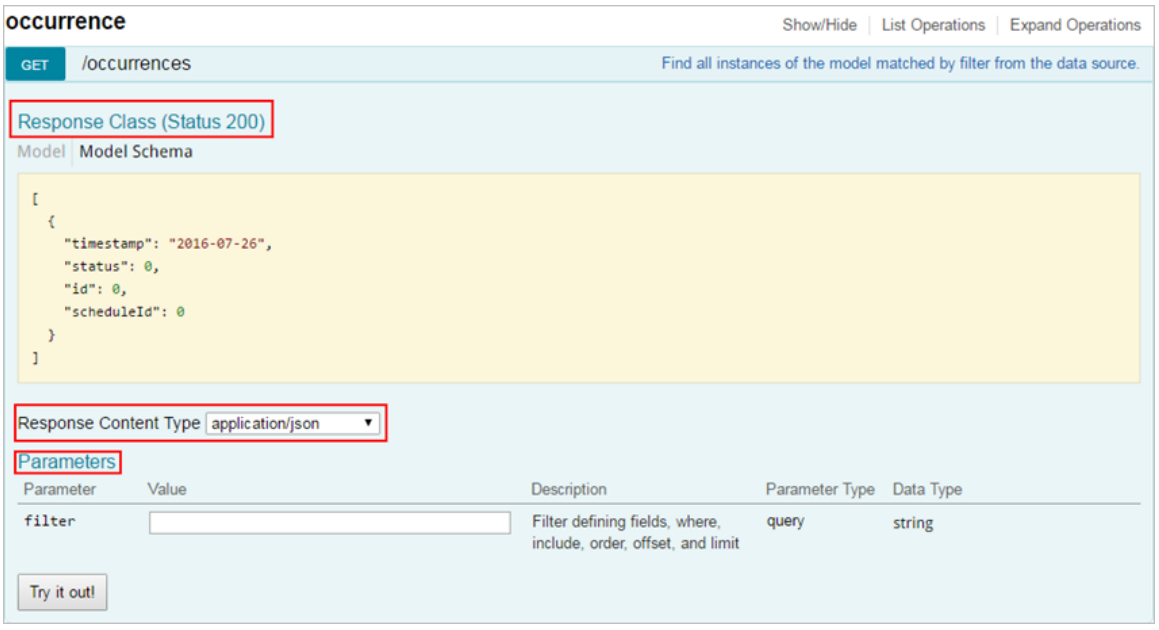


TABLE 3 API Dialog Box

Name	Description
Response Class	The Response interface of the API represents the response to a request.
Response Content Type	Content-type: application/json; designates the content to be in JSON format. This is the default type.

TABLE 3 API Dialog Box (continued)

Name	Description
Parameters	
Parameter	Use the filter parameter to supply a dimension you want to filter on, followed by the filter expression.
Value	The Parameter Value contains the value to be included in the request.
Description	Auto displays the parameter description
Parameter Type	Lists the API parameter types that you can use in the path or query parameters for your backend API methods, and the types you can use as method return types or request body types.
Data Type	Lists the data types used in API parameters and notification message fields.
Try it out	Click on this icon to view the response body based on the parameters set.
Curl	Curl is a command line tool and library for transferring data with URL syntax. Use the curl command to simulate HTTP verbs such as HEAD, GET, POST, PUT and DELETE request calls to the API.
Request URL	Contains the URL of the response.
Response Body	The response interface represents the response to a request.
Response Code	Contains the status code of the response. For example, 200 for a success.
Response Header	Contains the headers object associated with the response.

Using SCI API Explorer

After you log in to the SCI User Interface, you can access the SCI API Explorer to generate reports by using a variety of APIs that SCI supports.

This section provides steps on how to enter the SCI API Explorer to log in, after which the following tasks are shown:

- Generating the Access Token on page 13
- Querying to Obtain Report IDs on page 15
- Querying to Obtain Section IDs of a Specific Report on page 18
- Querying the Data Endpoint on page 20
- Logging in to API Explorer Programmatically on page 23

1. Enter `https://<your SCI IP address>/explorer` in your web browser.

2. Log in using the following default credentials:

   Username: admin

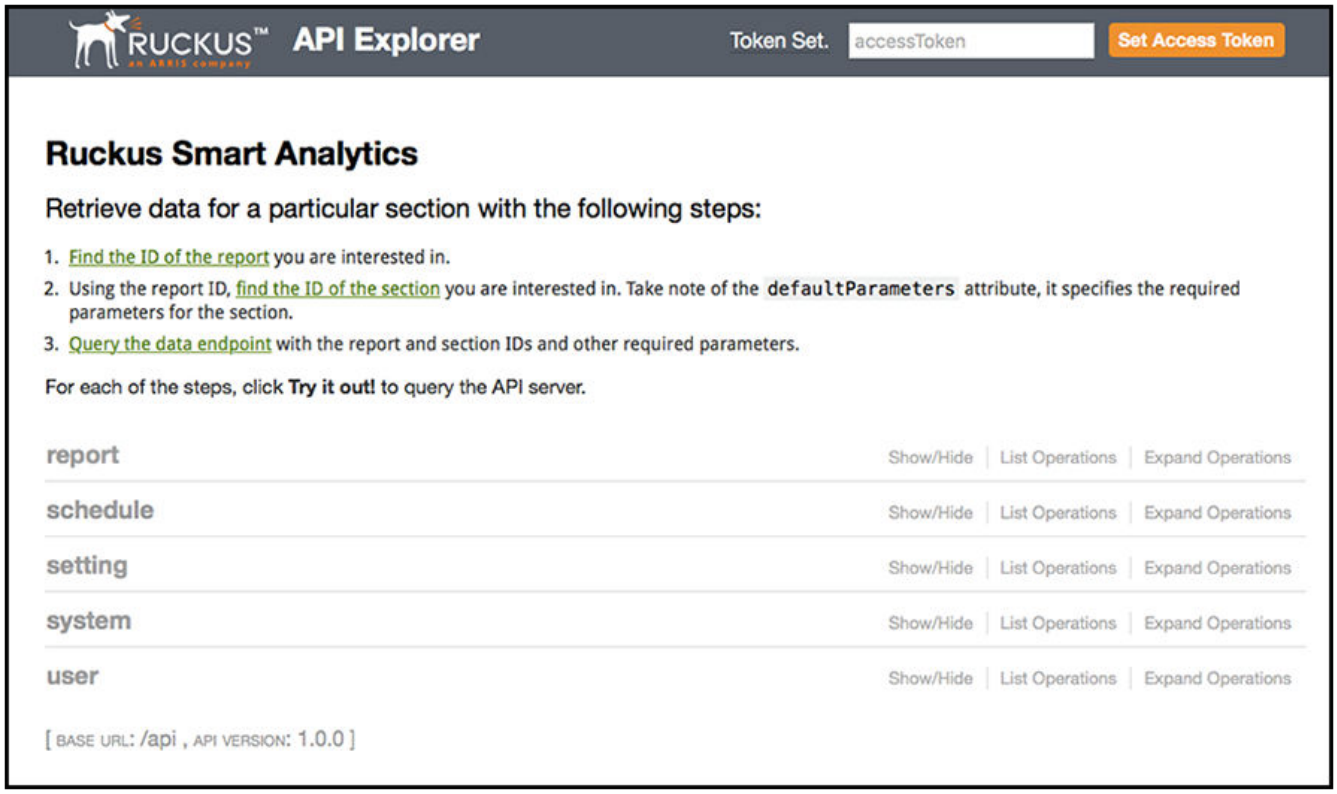
   Password: password

   NOTE

   If you have changed the default password, please use the new password accordingly.

   The following screen appears:

FIGURE 4 Ruckus Smart Analytics Screen when Entering API Explorer



This screen exposes all the APIs from the SCI core engine.

Generating the Access Token

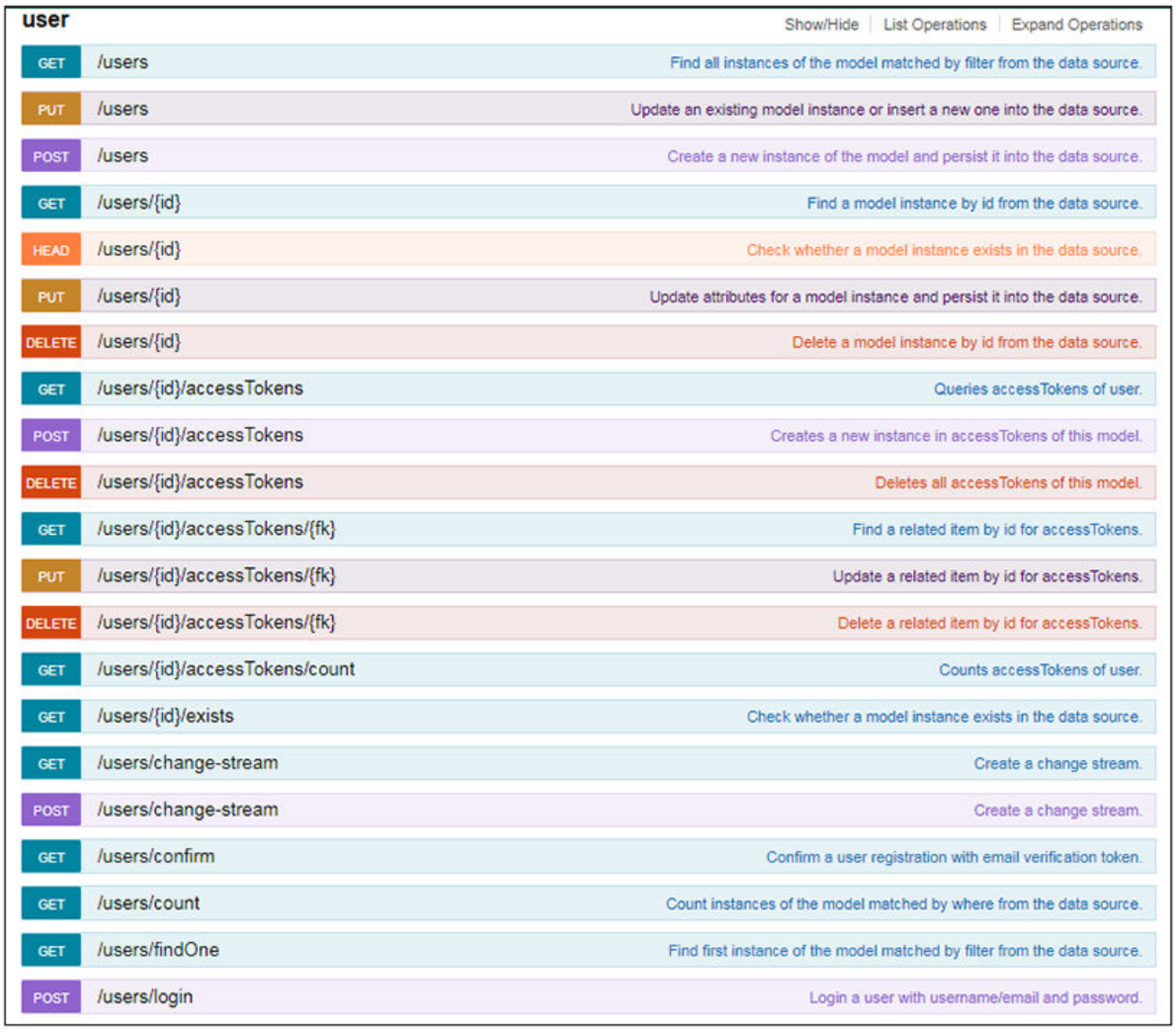
Before you can access all the APIs, you need to generate an access token.

Follow the steps below to generate the API access token:

1. Click **User** on the screen shown above.

   The screen expands as follows:

   FIGURE 5 Clicking User



2. Click on POST /users/login on the screen shown above.

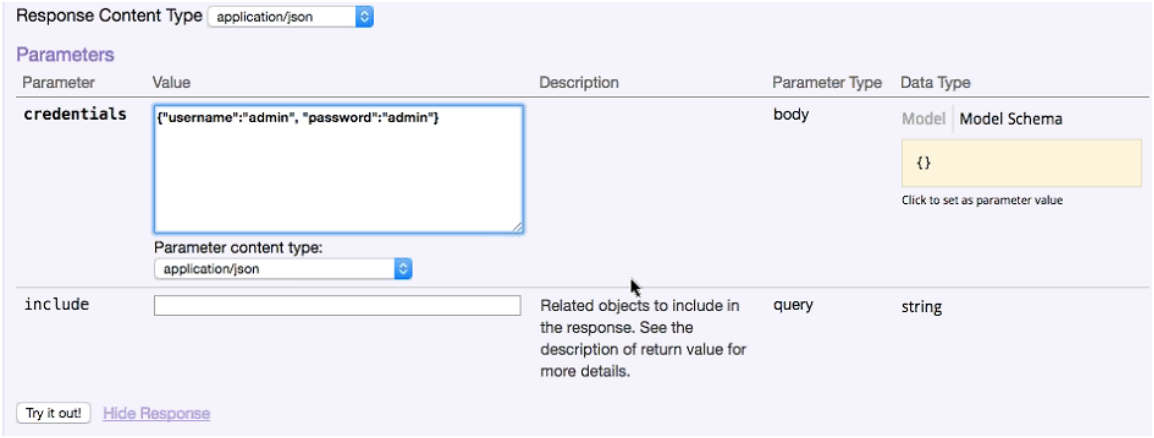
3. In the **credentials** section, enter the user credentials you used to log into the user interface, which, by default, are:

   Username: admin

   Password: password

   Input the string in the format **{"username":"admin", "password":"password"}** as shown below:

   FIGURE 6 Credentials



4. Click the **Try it out!** tab in the Credentials section.

   The access code you need is generated in the Response body of the curl script, as displayed below:

   FIGURE 7 Access Token

5. Copy this access token *(without the quotation marks*) and paste it in the **Set Access Token** field displayed at the top of the SCI user interface.

6. Click **Set Access Token**.

   If successful, the "Token Set" string displays to the left of the space where you pasted in the token:

   FIGURE 8 Token Set

   NOTE
   Now you can access all the API reports in the system. You might not get the whole report in the Response Body if it is a large amount of data. You can access the reports by using:

   - A Request URL. Paste this URL in a web browser to access the reports.
   - A curl script to SSH on to your machine and direct those reports to a desired location.

Querying to Obtain Report IDs

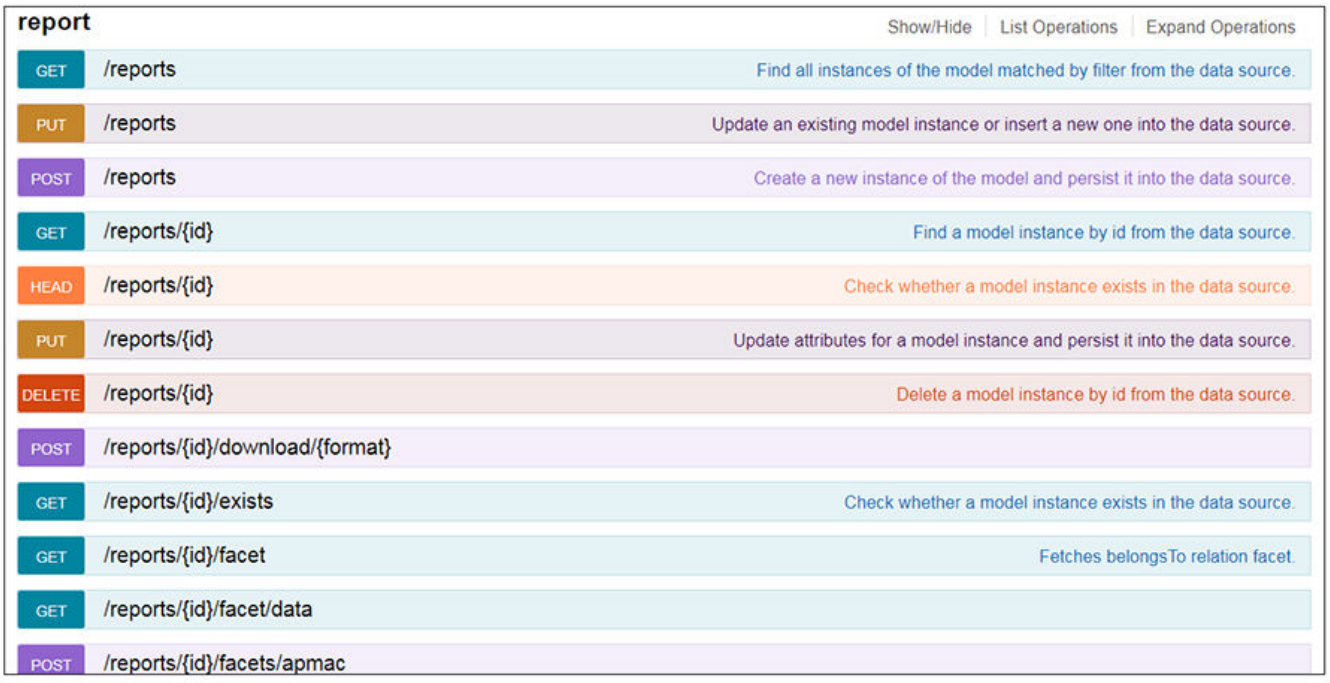
Each report in the SCI user interface is associated with a unique Report ID in the API. Each report consequently contains unique section IDs for each section within a report. This will also be shown in the following examples.

Follow the steps below to determine the list of IDs for each report.

1. Click **Report** on the Ruckus Smart Analytics screen.

   You should then get a display that includes the following:

   FIGURE 9 Display after clicking on Report



2. Click **GET/reports**.

   You should then get a display such as the following:

   FIGURE 10 GET/reports Response

3. Click **Try it out!**

   Check the output in the Response Body. The following is a portion of that output:


```
[
  {
    "title": "Clients Report",
    "urlSegmentName": "clients",
    "filterDataSource": "binnedSessions",
    "excludedFilters": null,
    "layout": [
      {
        "desiredWidth": "full",
        "layout": [
          {
            "section": 12,
            "desiredWidth": "half"
          },
          {
            "section": 13,
            "desiredWidth": "half"
          }
        ]
      },
      {
        "desiredWidth": "full",
        "section": 14
      },
      {
        "desiredWidth": "full",
        "section": 15
      },
      {
        "desiredWidth": "full",
        "section": 16
      }
    ],
    "headers": [
      "reportFilter",
      "periodButton",
      "savedFilters",
      "downloadButton"
    ],
    "routeParameters": null,
    "id": 1,
    "facetId": null
  },
```


   The code block shown above is for the "Clients Report," as you can see by the title in the first line of the code block. The Report ID is 1, which you can see near the end of the code block. The ID is always shown at the *end* of the corresponding report. Similarly, you can scroll through the output in the Response Body to obtain all the report IDs. For example, you will find that "Network Report" is ID 2.

Querying to Obtain Section IDs of a Specific Report

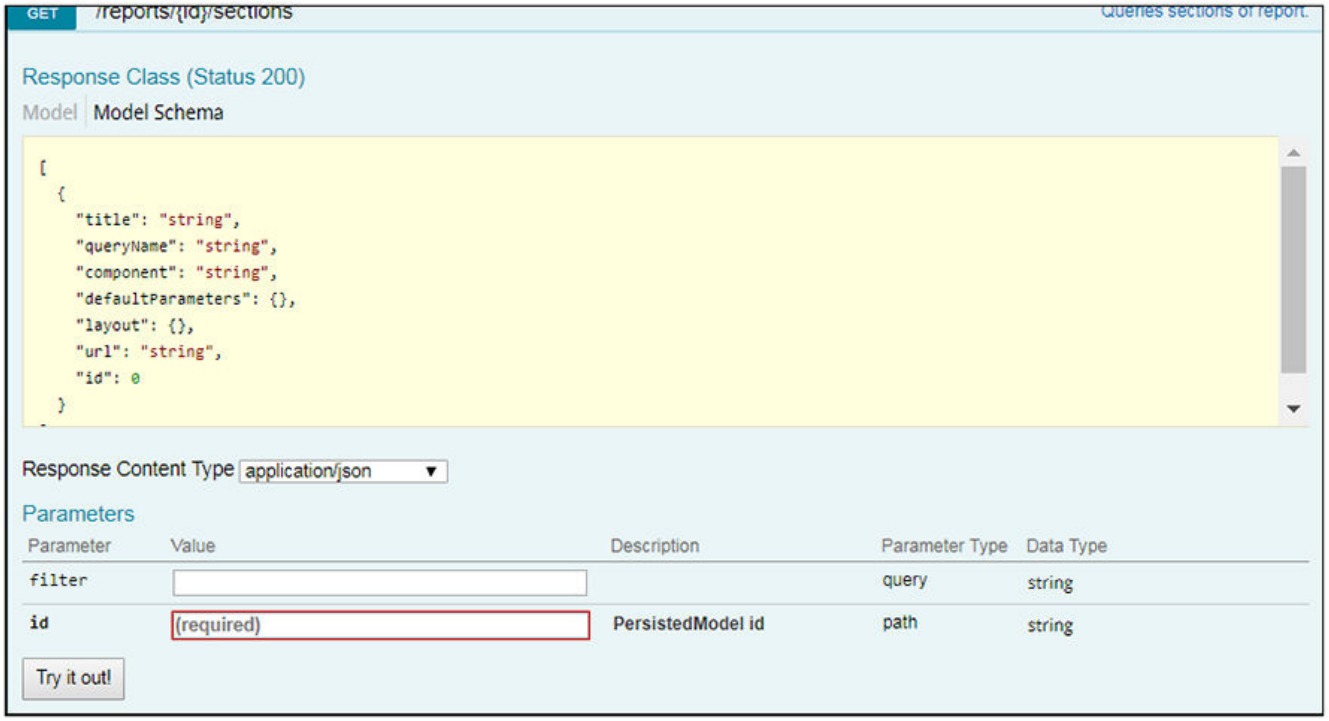
Each report in SCI contains multiple sections. Once you know the report IDs, you can obtain the names of each section within a report and their corresponding, unique section IDs.

Follow these steps to obtain the section titles and corresponding section IDs for all sections in a given report.

1. Once you know the ID of the report you want, click **GET/reports/{id}/sections**.

   The following is displayed.

   FIGURE 11 Display from clicking GET/reports/{id}/sections



2. Enter the Report ID in the "id" field. For example, for the Clients Report, enter an ID of 1.

3. Click **Try it out!**

A portion of the Response Body output is shown below:


```
[
  {
    "title": "Overview",
    "queryName": "overview",
    "component": "ReportOverview",
    "defaultParameters": {
      "granularity": "all"
    },
    "layout": {
      "width": "half",
      "widgetTheme": "blue"
    },
    "url": null,
    "id": 12
  },
  {
    "title": "Top 10 Unique Clients by Traffic",
    "queryName": "topChart",
    "component": "BarChart",
    "defaultParameters": {
      "granularity": "all",
      "metric": "traffic"
    },
    "layout": {
      "width": "half",
      "headers": [
        {
          "component": "SelectFilter",
          "name": "metric",
          "options": {
            "traffic": "User Traffic",
            "rxBytes": "Rx User",
            "txBytes": "Tx User"
          }
        }
      ],
      "format": "bytesFormat",
      "colors": [
        "#5BA1E0",
        "#5BA1E0",
        "#5BA1E0",
        "#76CEF5",
        "#76CEF5",
        "#76CEF5",
        "#D9E6F5",
        "#D9E6F5",
        "#D9E6F5",
        "#D9E6F5"
      ],
      "drillDownRoute": "/report/client/${x}"
    },
    "url": null,
    "id": 13
  },
```


The code block shown above is for the sections of the "Clients Report." The first segment of the Response Body above shows the title of "Overview." If you scroll down to the end of that segment, you see that the ID for that section is 12. The ID of a section is always shown at the *end* of the corresponding segment for the section. The ID for the next section shown above, "Top 10 Unique Clients by Traffic", is 13. You can scroll through the output in the Response Body to obtain all the section IDs for the report you have identified.

Querying the Data Endpoint

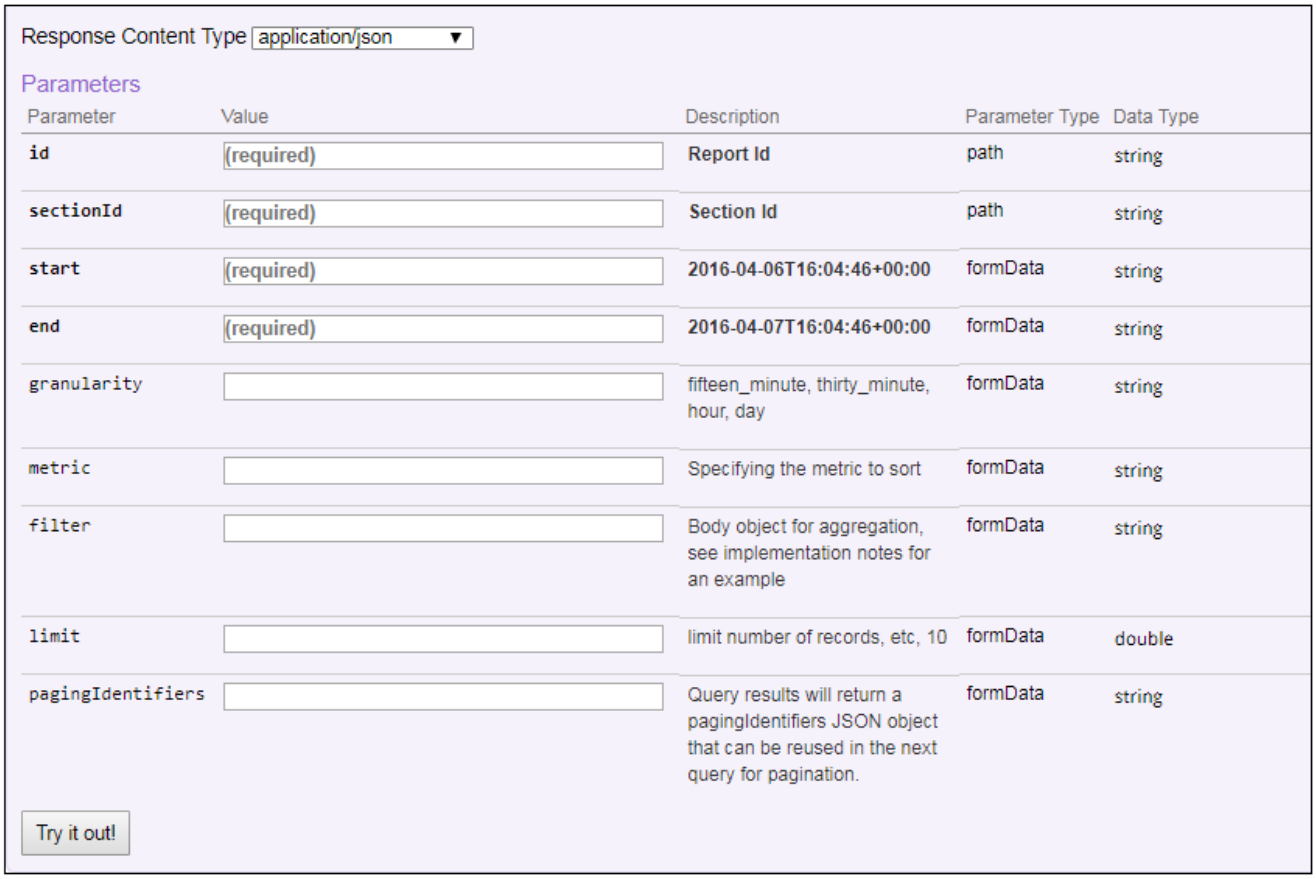
Once you know the Report ID and the Section ID you are interested in, you can query for specific data.

Follow these steps to query for specific data based on the Report ID and Section IDs.

1. Click **POST/reports/{id}/sections/{sectionId}/data**.

   The following parameters are displayed.

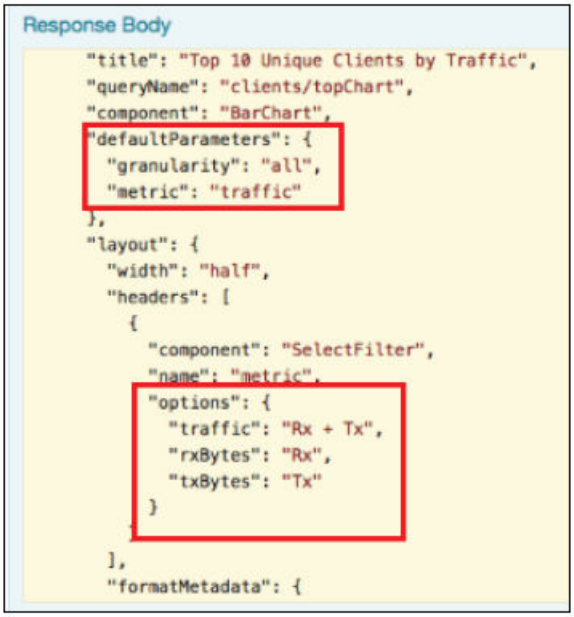
   FIGURE 12 Display from clicking POST/reports/{id}/sections/{sectionId}/data



2. Fill out the required fields, using the Report ID and Section ID that you have already determined from the previous queries described above.

   To determine if the metric parameter is required, look at the Response Body from for the section you are interested in, and check if "metric" is found under "defaultParameters". If it is, enter one of the values listed under "options." The figure below illustrates this scenario.

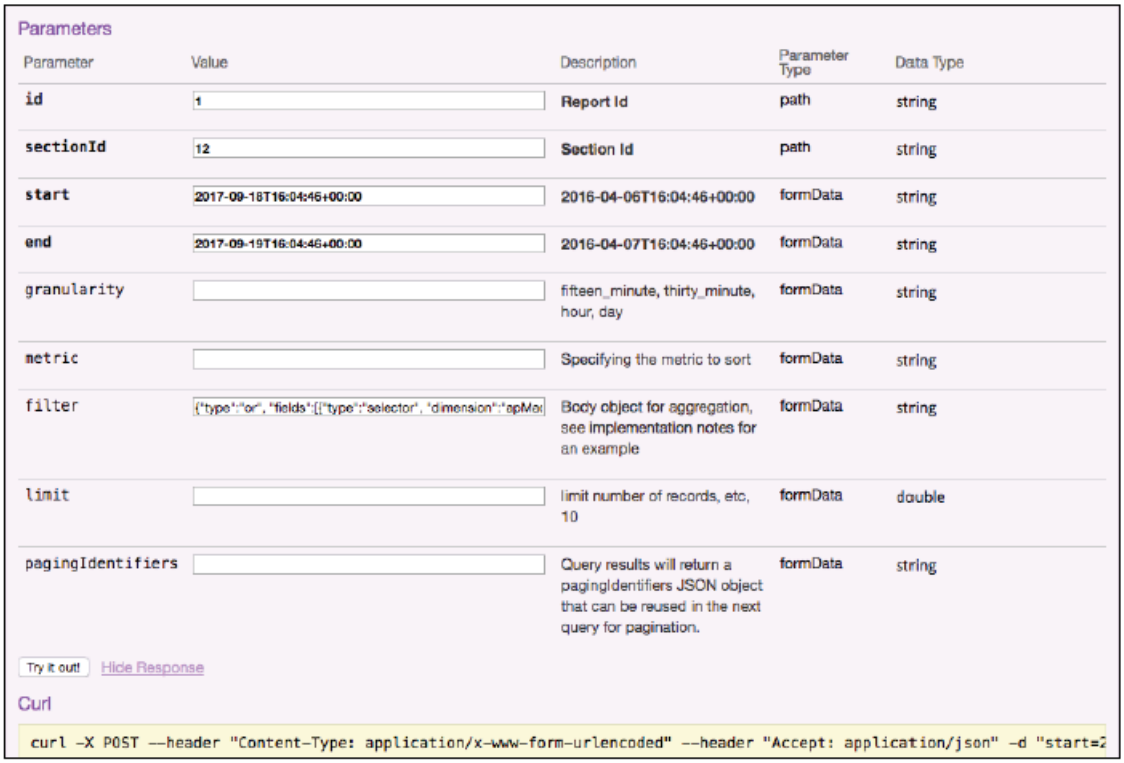
   FIGURE 13 Response Body that indicates a Metric is required



3. You can also use the optional parameters. An example would be filtering on an AP MAC address of: 00:AA:BB:CC:44:D0. To do this, you would enter this exact string shown below into the filter field in the screen above: {"type":"or", "fields":[{"type":"selector", "dimension":"apMac", "value":"00:AA:BB:CC:44:D0"}]}

   If you want to generate a report using this filter, the Report ID of 1 (Clients Report), Section ID 12 (Overview section of Clients Report), and a time interval of your choice, the parameters once you have entered the information would appear as follows:

FIGURE 14 Parameter Example for Querying Data on a Specific Section ID



Other values you can use for dimensions in the "filter" string are shown below:

TABLE 4 Dimensions to use in filter string parameter

Dimension portion of filter string	Value to use for Dimension
System Name	"system"
Controller MAC	"ctrlMac"
Controller Name	"ctrlName"
Controller Serial	"ctrlSerial"
Zone	"zoneName"
AP Group	"apGroup"
AP MAC	"apMAC"
AP Name	"apName"
AP Serial	"apSerial"
SSID	"ssid"
Radio	"radio"

TABLE 4 Dimensions to use in filter string parameter (continued)

Dimension portion of filter string	Value to use for Dimension
Session Type	"sessionType"

4. **Click Try it out!**.

   The data output is displayed in the Response Body.

Logging in to API Explorer Programmatically

You can log in using the curl command for POST /users/login. Ensure you use the –k option in the curl command.

The response to the call will contain a token that you can use in subsequent calls.

In subsequent calls, you can pass this token as a header “Authorization:<token>”.

The following example shows how to query section 28 for report ID 7 programmatically using the curl command.

```
curl -k -X POST --header "Content-Type: application/x-www-form-urlencoded" --header "Authorization: <your-
auth-token>" --header "Accept:
application/json" -d "start=2016-12-13T20%3A30%3A46%2B00%3A00&end=2016-12-13T20%3A45%3A46%2B00%3A00"
"https://<your-SCI-I
P>/api/reports/7/sections/28/data"
```

Ruckus Smart Analytics

Retrieve data for a particular section with the following steps:

1. Find the ID of the report you are interested in.
2. Using the report ID, find the ID of the section you are interested in. Take note of the `defaultParameters` attribute, it specifies the required parameters for the section.
3. Query the data endpoint with the report and section IDs and other required parameters.

For each of the steps, click **Try it out!** to query the API server.

alert

POST /alerts/sendNotification

Response Class (Status 200)

Model | Model Schema

```
{}
```

Response Content Type application/json

Try it out!

---

report

GET /reports — Find all instances of the model matched by filter from the data source.

Response Class (Status 200)

Model | Model Schema


```
[
  {
    "title": "string",
    "urlSegmentName": "string",
    "filterDataSource": "string",
    "component": "string",
    "excludedFilters": [
      {}
    ],
    "layout": [
      {}
    ],
    "headers": [
      {}
    ],
    "routeParameters": {},
    "datasourcesUsed": [
      {}
    ],
    "id": 0
```

```
  }
]
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter		Filter defining fields, where, include, order, offset, and limit	query	string

Try it out!

GET /reports/{id}

Find a model instance by id from the data source.

Response Class (Status 200)

Model | Model Schema


```
{
  "title": "string",
  "urlSegmentName": "string",
  "filterDataSource": "string",
  "component": "string",
  "excludedFilters": [
    {}
  ],
  "layout": [
    {}
  ],
  "headers": [
    {}
  ],
  "routeParameters": {},
  "datasourcesUsed": [
    {}
  ],
  "id": 0
}
```


Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	**Model id**	path	string
filter		Filter defining fields and include	query	string

Try it out!

POST /reports/{id}/download/{format}

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	**report id**	path	string
format	(required)	**pdf or csv**	path	string
state	(required)	**page state**	formData	string
timezone	(required)	**time zone identifier**	formData	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Request was successful		

Try it out!

GET /reports/{id}/sections — Queries sections of report.

Response Class (Status 200)

Model | Model Schema


```
[
  {
    "title": "string",
    "queryName": "string",
    "systemOwnerOnly": false,
    "component": "string",
    "defaultParameters": {},
    "layout": {},
    "url": "string",
    "id": 0
  }
]
```


Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter			query	string
id	(required)	**PersistedModel id**	path	string

Try it out!

POST /reports/{id}/sections/{sectionId}/data

Implementation Notes

For the **filter** field below, an example would be

{ "type": "or", "fields": [{ "type": "selector", "dimension": "apMac", "value": "00000000

Response Class (Status 200)

Model | Model Schema


```
{
  "data": [
    {}
  ],
  "metadata": {}
}
```


Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	**Report Id**	path	string
sectionId	(required)	**Section Id**	path	string
start	(required)	**2016-04-06T16:04:46+00:00**	formData	string
end	(required)	**2016-04-07T16:04:46+00:00**	formData	string
granularity		fifteen_minute, thirty_minute, hour, day	formData	string
metric		Specifying the metric to sort	formData	string
filter		Body object for aggregation, see implementation notes for an example	formData	string
switchFilter		Body object for aggregation, see implementation notes for an example	formData	string
limit		limit number of records, etc, 10	formData	double
pagingIdentifiers		Query results will return a pagingIdentifiers JSON object that can be reused in the next query for pagination.	formData	string

Try it out!

POST /reports/withRelations

Implementation Notes

For the **`urlSegmentName`** field below, examples could be `overview`, `network`, `ap`, `clients`

Response Class (Status 200)

Model | Model Schema

```
{}
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
urlSegmentName	(required)		formData	string

Try it out!

schedule

PUT /schedules/{id}/updateWithRelations

Update schedule with filter and occurrence in a single transaction.

Response Class (Status 200)

Model | Model Schema

```
{
  "name": "string",
  "format": "string",
  "frequency": "string",
  "day": 0,
  "hour": 0,
  "enabled": true,
  "timezone": "string",
  "recipients": [
    {}
  ],
  "id": 0,
  "userId": 0,
  "reportId": 0,
  "filterId": 0
}
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	**PersistedModel id**	path	string

Parameter	Value	Description	Parameter Type	Data Type
scheduleData	(required)	**JSON string for schedule (see POST /schedules)**	formData	string
filterData	(required)	**JSON string for filter (e.g. `{ filter: "compressed filter" }`)**	formData	string

Try it out!

POST /schedules/batchDelete

Delete schedules with their related filters and occurrences in a single transaction.

Response Class (Status 200)

Model | Model Schema

```
{
  "count": 0
}
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
ids	(required)	**Array of PersistedModel ids (e.g. `[1, 2, 3]`)**	formData	string

Try it out!

POST /schedules/createWithRelations

Create schedule with filter and occurrence in a single transaction.

Response Class (Status 200)

Model | Model Schema

```
{
  "name": "string",
  "format": "string",
  "frequency": "string",
  "day": 0,
  "hour": 0,
  "enabled": true,
  "timezone": "string",
  "recipients": [
    {}
  ],
  "id": 0,
  "userId": 0,
```

```
  "reportId": 0,
  "filterId": 0
}
```


Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
reportId	(required)	**Report Id**	formData	string
scheduleData	(required)	**JSON string for schedule (see POST /schedules)**	formData	string
filterData	(required)	**JSON string for filter (see POST /filters)**	formData	string

Try it out!

POST /schedules/executeJob — Run schedule job

Response Class (Status 200)

Model | Model Schema


```
{}
```


Response Content Type application/json

Try it out!

setting

PUT /settings — Update an existing model instance or insert a new one into the data source.

Response Class (Status 200)

Model | Model Schema


```
{
  "key": "string",
  "values": {}
}
```


Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data		Model instance data	body	Model | Model Schema

```
{
  "key": "string",
  "values": {}
}
```

Click to set as parameter value

Parameter content type: application/json

Try it out!

GET /settings/{id} — Find a model instance by id from the data source.

Response Class (Status 200)

Model | Model Schema

```
{
  "key": "string",
  "values": {}
}
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	**Model id**	path	string
filter		Filter defining fields and include	query	string

Try it out!

POST /settings/sendTestEmail

Response Class (Status 200)

Model | Model Schema

```
{}
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
recipients	(required)	**Comma-separated list of recipients**	formData	string

Try it out!

system

GET /systems — Find all instances of the model matched by filter from the data source.

Response Class (Status 200)

Model | Model Schema


```
[
  {
    "id": "string",
    "type": "string",
    "format": "string",
    "version": "v1",
    "pciData": {},
    "location": "string",
    "backupLocation": "",
    "user": "string",
    "lastContact": "2020-02-13T10:15:20.520Z"
  }
]
```


Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter		Filter defining fields, where, include, order, offset, and limit	query	string

Try it out!

POST /systems — Create a new instance of the model and persist it into the data source.

Response Class (Status 200)

Model | Model Schema


```
{
  "id": "string",
  "type": "string",
  "format": "string",
  "version": "v1",
  "pciData": {},
  "location": "string",
  "backupLocation": "",
  "user": "string",
  "lastContact": "2020-02-13T10:15:20.521Z"
}
```


Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
data		Model instance data	body	Model | Model Schema

Parameter content type: application/json


```
{
  "id": "string",
  "type": "string",
  "format": "string",
  "version": "v1",
  "pciData": {},
  "location": "string",
  "backupLocation": "",
  "user": "string",
  "lastContact": "2020-02-13T1
}
```


Click to set as parameter value

Try it out!

GET /systems/{id} Find a model instance by id from the data source.

Response Class (Status 200)

Model | Model Schema


```
{
  "id": "string",
  "type": "string",
  "format": "string",
  "version": "v1",
  "pciData": {},
  "location": "string",
  "backupLocation": "",
  "user": "string",
  "lastContact": "2020-02-13T10:15:20.524Z"
}
```


Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	**Model id**	path	string
filter		Filter defining fields and include	query	string

Try it out!

PUT /systems/{id} Update attributes for a model instance and persist it into the data source.

Response Class (Status 200)

Model | Model Schema

```
{
  "id": "string",
  "type": "string",
  "format": "string",
  "version": "v1",
  "pciData": {},
  "location": "string",
  "backupLocation": "",
  "user": "string",
  "lastContact": "2020-02-13T10:15:20.526Z"
}
```


Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	Parameter content type: application/json	An object of model property name/value pairs	body	Model | Model Schema { "id": "string", "type": "string", "format": "string", "version": "v1", "pciData": {}, "location": "string", "backupLocation": "", "user": "string", "lastContact": "2020-02- } Click to set as parameter value
id	**(required)**	**PersistedModel id**	path	string

Try it out!

DELETE /systems/{id} Delete a model instance by id from the data source.

Response Class (Status 200)

Model | Model Schema


```
{}
```


Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	**(required)**	**Model id**	path	string

Try it out!

user

GET /users — Get users that current user can manage.

Response Class (Status 200)

Model | Model Schema


```
[
  {
    "firstName": "string",
    "lastName": "string",
    "lastLogin": "2020-02-13T10:15:20.530Z",
    "isExternal": false,
    "username": "string",
    "email": "string",
    "id": 0
  }
]
```


Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter		Filter defining fields, where, include, order, offset, and limit	query	string

Try it out!

GET /users/{id} — Find a model instance by id from the data source.

Response Class (Status 200)

Model | Model Schema


```
{
  "firstName": "string",
  "lastName": "string",
  "lastLogin": "2020-02-13T10:15:20.532Z",
  "isExternal": false,
  "username": "string",
  "email": "string",
  "id": 0
}
```


Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	**Model id**	path	string

Parameter	Value	Description	Parameter Type	Data Type
filter		Filter defining fields and include	query	string

Try it out!

GET /users/{id}/filters — Queries filters of user.

Response Class (Status 200)

Model | Model Schema


```
[
  {
    "name": "string",
    "urlSegmentName": "string",
    "filter": "string",
    "id": 0,
    "reportId": 0,
    "userId": 0
  }
]
```


Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter			query	string
id	(required)	**User id**	path	string

Try it out!

GET /users/{id}/filters/{fk} — Find a related item by id for filters.

Response Class (Status 200)

Model | Model Schema


```
{
  "name": "string",
  "urlSegmentName": "string",
  "filter": "string",
  "id": 0,
  "reportId": 0,
  "userId": 0
}
```


Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
fk	(required)	**Foreign key for filters**	path	string
id	(required)	**User id**	path	string

Try it out!

GET /users/{id}/schedules — Queries schedules of user.

Response Class (Status 200)

Model | Model Schema


```
[
  {
    "name": "string",
    "format": "string",
    "frequency": "string",
    "day": 0,
    "hour": 0,
    "enabled": true,
    "timezone": "string",
    "recipients": [
      {}
    ],
    "id": 0,
    "userId": 0,
    "reportId": 0,
    "filterId": 0
  }
]
```


Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter			query	string
id	(required)	**User id**	path	string

Try it out!

POST /users/login — Login a user with username/email and password.

Response Class (Status 200)

Model | Model Schema


```
{}
```


Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
credentials	(required) Parameter content type: application/json		body	Model Model Schema {} Click to set as parameter value
include		Related objects to include in the response. See the description of return value for more details.	query	string

Try it out!

POST /users/logout — Logout a user with access token.

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Request was successful		

Try it out!

[BASE URL: /api , API VERSION: 1.0.0]

COMMSCOPE®
RUCKUS®

© 2020 CommScope, Inc. All rights reserved.
Ruckus Wireless, Inc., a wholly owned subsidiary of CommScope, Inc.
350 West Java Dr., Sunnyvale, CA 94089 USA
www.ruckuswireless.com