



# Ruckus Wireless™ SmartCell Insight™

## API User Guide for Release 2.5.1

Part Number: 800-71484-001 Rev A  
Published: 30 March 2017

[www.ruckuswireless.com](http://www.ruckuswireless.com)

# Contents

Copyright Notice and Proprietary Information.....	3
About This Guide.....	4
Document Conventions.....	4
Related Documentation.....	5
Documentation Feedback.....	5
<b>1 SmartCell Insight Overview</b>	
Accessing the API.....	6
Navigating to the UI.....	6
Generating Access Token.....	7
Using the API Dialog Box.....	9
<b>2 Report Types</b>	
Common Tasks.....	13
Parameters for Generating API Reports.....	14
Accessing and Retrieving Data.....	16
Find the ID of the Report.....	16
Find the ID of the Section.....	17
Query the Data Endpoint.....	18
Generating API Reports.....	19
<b>3 Ruckus Smart Analytics</b>	

# Copyright Notice and Proprietary Information

Copyright 2017. Ruckus Wireless, Inc. All rights reserved.

No part of this documentation may be used, reproduced, transmitted, or translated, in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without prior written permission of Ruckus Wireless, Inc. ("Ruckus"), or as expressly provided by under license from Ruckus.

## **Destination Control Statement**

Technical data contained in this publication may be subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

## **Disclaimer**

THIS DOCUMENTATION AND ALL INFORMATION CONTAINED HEREIN ("MATERIAL") IS PROVIDED FOR GENERAL INFORMATION PURPOSES ONLY. RUCKUS AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THE MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, OR THAT THE MATERIAL IS ERROR-FREE, ACCURATE OR RELIABLE. RUCKUS RESERVES THE RIGHT TO MAKE CHANGES OR UPDATES TO THE MATERIAL AT ANY TIME.

## **Limitation of Liability**

IN NO EVENT SHALL RUCKUS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES, OR DAMAGES FOR LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY YOU OR ANY THIRD PARTY, WHETHER IN AN ACTION IN CONTRACT OR TORT, ARISING FROM YOUR ACCESS TO, OR USE OF, THE MATERIAL.

## **Trademarks**

Ruckus Wireless, Ruckus, the bark logo, BeamFlex, ChannelFly, Dynamic PSK, FlexMaster, Simply Better Wireless, SmartCell, SmartMesh, SmartZone, Unleashed, ZoneDirector and ZoneFlex are trademarks of Ruckus Wireless, Inc. in the United States and other countries. All other product or company names may be trademarks of their respective owners.

# 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

[Table 1: Text conventions](#) on page 4 and [Table 2: Notice conventions](#) on page 5 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
<b>user interface controls</b>	Keyboard keys, software buttons, and field names	Click <b>Create New</b>
<b>Start &gt; All Programs</b>	Represents a series of commands, or menus and submenus	Select <b>Start &gt; All Programs</b>
<b>ctrl+V</b>	Represents keyboard keys pressed in combination	Press <b>ctrl+V</b> to paste the text from the clipboard.
<b>screen or page names</b>		Click <b>Advanced Settings</b> . The <b>Advanced Settings</b> page appears.
<b>command name</b>	Represents CLI commands	

Convention	Description	Example
parameter name	Represents a parameter in a CLI command or UI feature	
<i>variable name</i>	Represents variable data	<i>{ZoneDirectorID}</i>
filepath	Represents file names or URI strings	http://ruckuswireless.com

Table 2: Notice conventions

Notice type	Description
<b>NOTE</b>	Information that describes important features or instructions
<b>CAUTION!</b>	Information that alerts you to potential loss of data or potential damage to an application, system, or device
<b>WARNING!</b>	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](mailto:docs@ruckuswireless.com)

When contacting us, please include the following information:

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

# SmartCell Insight Overview

# 1

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.

---

**NOTE** Refer to the Ruckus Smart Analytics Appendix on using each of the APIs.

---

## Accessing the API

You can access the API after you log into the SCI User Interface.

You can access most of the reports mentioned in this section using the user interface, but if you require a detailed report at a granular level you can query the API which pulls the data from the core engine. Also the user interface limits the reports to top 100, if you require more than that you can access the API and procure them.

## Navigating to the UI

To log into the SCI User Interface, use the following URL and the credentials.

1. Enter `https://<your SCI IP address>/explorer` in your web browser.
2. Log in using the following default credentials:

Username: admin

Password: admin

This page exposes all the APIs from the SCI core engine. If you have changed the default password, please use the new password accordingly.

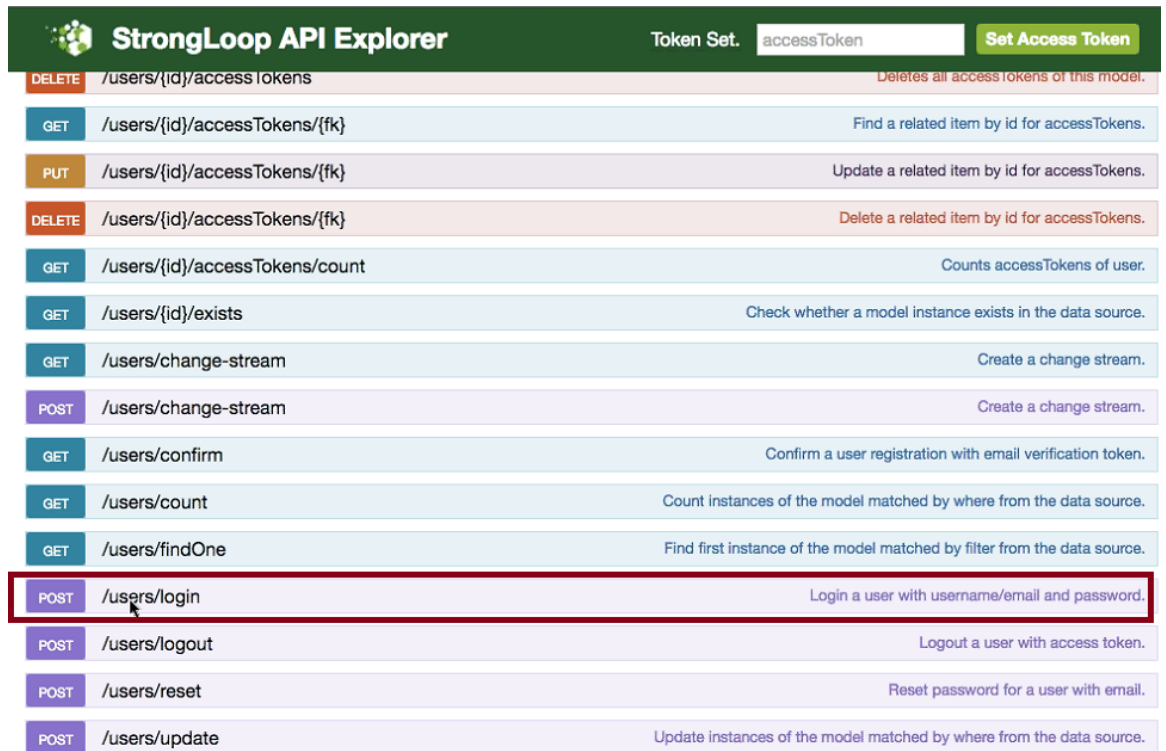
## Generating Access Token

You can access these APIs after you generate an access token.

To generate the access token:

1. Go to the section **Users > Login**. The following screenshot illustrates the same.

Figure 1: User Login



2. In the **credentials** section, enter the user credentials you used to log into the user interface.

Username: admin

Password: admin

Input a string in the format `{"username":"admin", "password":"admin"}`

3. Click the **Try it out!** tab

Figure 2: Credentials

## SmartCell Insight Overview

Accessing the API

The screenshot shows the 'Parameters' section of an API Explorer. At the top, 'Response Content Type' is set to 'application/json'. Below it, a table lists parameters:

Parameter	Value	Description	Parameter Type	Data Type
credentials	<pre>{\"username\":\"admin\", \"password\":\"admin\"}</pre>		body	Model Model Schema <pre>{}</pre> Click to set as parameter value
include	<input type="text"/>	Related objects to include in the response. See the description of return value for more details.	query	string

At the bottom, there are buttons for 'Try it out!' and 'Hide Response'.

You will get your access token in the **Response body** of the curl script as displayed below.

Figure 3: Access Token

The screenshot shows the 'Curl' section with the following command:

```
curl -X POST --header "Content-Type: application/json" --header "Accept: application/json" -d "{\"username\":\"rsa\", \"password\":\"rsa\"}"
```

The 'Request URL' section shows:

```
https://rsa-staging.ruckuslbs.com/api/users/login?access_token=Jn4mgkNl8dIEFAAR4nvtwGFJ6KuCedGTxZ5uZFnSS0ZdLFJ3gApGbWj3
```

The 'Response Body' section shows the following JSON response:

```
{  
  "id": "tn33XDf40CsAiWmPOhXUu3I1KNUTwtUVlf8MJ9aUwmqoIZTzimahD9GNvLD0RbTb",  
  "ttl": 1209600,  
  "created": "2016-07-21T02:39:03.265Z",  
  "userId": 1  
}
```

4. Copy this access token and paste it in the **Token Set** field displayed at the top of the SCI user interface.
5. Click **Set Access Token**

Figure 4: Set Access Token



Now you can access all the API reports in the system.



---

**NOTE** You might not get the whole report in the Response Body if it is huge 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.
- 

## 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 5: API Dialog Box

The screenshot displays the API Dialog Box for the 'occurrence' endpoint. At the top, it shows the method 'GET' and the path '/occurrences'. Below this, there is a section for the 'Response Class (Status 200)' which includes a 'Model Schema' tab and a JSON response body. The response body is a list containing one object with fields: 'timestamp' (2016-07-26), 'status' (0), 'id' (0), and 'scheduleId' (0). Below the response body, there is a 'Response Content Type' dropdown menu set to 'application/json'. At the bottom, there is a 'Parameters' section with a table:

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>	Filter defining fields, where, include, order, offset, and limit	query	string

A 'Try it out!' button is located at the bottom left of the 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.
<b>Parameters</b>	
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.



# Report Types

# 2

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

## **Occurrence**

This provides the API report related to the retrieval of occurrence records based on the scheduler. (Refer to the SCI User Guide for details).

## **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.

## **ZDXML**

This report type is Zone Director's specific calls.

---

**NOTE** Refer to the **Ruckus Smart Analytics Appendix** on using each of the report types.

---

## 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 tasks, which is common .

### 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 6: Common tasks

<b>occurrence</b>	Show/Hide	List Operations	Expand Operations
<b>report</b>	Show/Hide	List Operations	Expand Operations
<b>schedule</b>	Show/Hide	List Operations	Expand Operations
<b>setting</b>	Show/Hide	List Operations	Expand Operations
<b>system</b>	Show/Hide	List Operations	Expand Operations
<b>user</b>	Show/Hide	List Operations	Expand Operations
<b>zdXml</b>	Show/Hide	List Operations	Expand Operations

## Report Types

Parameters for Generating API Reports

# Parameters for Generating API Reports

This section lists the parameters available to generate a report pertaining to occurrence. Similarly, an array of parameters are available for different report types.

The table below contains a sample of the parameters available in generating a report pertaining to occurrence.

Table 4: Parameter Details

Resource URL	Method	Response	Description
/occurrence	GET	Timestamp Status ID Schedule ID	Finds all instances of the model matched by filter from the data source.
/occurrences	PUT	Timestamp Status ID Schedule ID	Updates an existing model instance or inserts a new model into the data source.
/occurrences	POST	Timestamp Status ID Schedule ID	Creates a new instance of the model and persist it into the data source.
/occurrences/{id}	GET	Timestamp Status ID Schedule ID	Finds a model instance by identifier from the data source.
/occurrences/{id}	HEAD	Exists	Checks if a model instance exists in the data source.
/occurrences/{id}	PUT	Timestamp Status ID Schedule ID	Updates attributes for a model instance and persist it into the data source.

Resource URL	Method	Response	Description
/occurrences/{id}	DELETE		Deletes a model instance by identifier from the data source.
/occurrences/{id}/exists	GET	Exists	Checks whether a model instance exists in the data source.
/occurrences/{id}/schedule	GET	Name Filter Format Frequency Day Hour Enabled Timezone Recipients ID Report ID	Retrieves the relation schedule.
/occurrences/change-stream	GET		Finds a change stream.
/occurrences/change-stream	POST		Creates a change stream.
/occurrences/count	GET	Count	Counts the instances of models matched based on data source.
/occurrences/findOne	GET	Timestamp Status ID Schedule ID	Finds the first instance of the model matched by filter from the data source.
/occurrences/update	POST		Updates instances of the models matched based on the data source.

## Report Types

Accessing and Retrieving Data

# Accessing and Retrieving Data

You can access and retrieve data for different sections of the Ruckus Smart Analytics. This section contains the available options and how to use them.

The options are:

- [Find the ID of the report](#)
- [Find the ID of the section](#)
- [Query the data endpoint](#)

## Find the ID of the Report

You can click on this link to search for a report using the ID.

Figure 7: Report

**report** Show/Hide | List Operations | Expand Operations

**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",
    "id": 0,
    "facetId": 0
  }
]
```

Response Content Type: application/json

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>	Filter defining fields, where, include, order, offset, and limit	query	string

[Try it out!](#)



## Find the ID of the Section

You can search using the report ID you are interested in. Take note of the default Parameters attribute, it specifies the required parameters for the section.

You can browse to the section for which you want the report using this link.

**Example:** If you enter 3 at the ID you get the respective information in the **Response Content Type** section

Figure 8: Search by ID of the Section

Response Class (Status 200)

Model | Model Schema

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

Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>		query	string
id	3 <input type="text"/>	PersistedModel id	path	string

## Report Types

Accessing and Retrieving Data

## Query the Data Endpoint

You can query the data endpoint with the report ID, section ID and other required parameters.

Figure 9: Query Data Endpoint

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

```
{ "type": "or", "fields": [{ "type": "selector", "dimension": "apMac", "value": "000000000000" }]}
```

**Response Class (Status 200)**  
Model | Model Schema

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

Response Content Type

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="(required)"/>	Report Id	path	string
sectionId	<input type="text" value="(required)"/>	Section Id	path	string
start	<input type="text" value="(required)"/>	2016-04-06T16:04:46+00:00	formData	string
end	<input type="text" value="(required)"/>	2016-04-07T16:04:46+00:00	formData	string
granularity	<input type="text"/>	fifteen_minute, thirty_minute, hour, day	formData	string

---

### NOTE

You have to specify the mandatory parameters such as id, sectionId, start, and end as string values.

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

## Generating API Reports

Based on report type you can generate various reports.

The following is an illustration of generating reports based on the report identifier.

1. Navigate to the **URL - GET /reports**
2. Click **Try it out!** which displays all the reports along with its report identifier
3. Navigate to the **GET /reports/{id}**
4. Enter the parameter identifier.
5. Click **Try it out!** which displays the reports pertaining to the identifier
6. Use the curl command or the request URL to view the report.

# 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.

## occurrence

GET	/occurrences	Find all instances of the model matched by filter from the data source.
<b>Response Class (Status 200)</b>		
Model   Model Schema		
<pre>[   {     "timestamp": "2016-07-12",     "status": 0,     "id": 0,     "scheduleId": 0   } ]</pre>		
Response Content Type	<input type="text" value="application/json"/>	

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>	Filter defining fields, where, include, order, offset, and limit	query	string

Try it out!

PUT /occurrences

Update an existing model instance or insert a new one into the data source.

## Response Class (Status 200)

Model | Model Schema

```
{
  "timestamp": "2016-07-12",
  "status": 0,
  "id": 0,
  "scheduleId": 0
}
```

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	<input type="text"/>	Model instance data	body	Model   Model Schema

```
{
  "timestamp": "2016-07-12",
  "status": 0,
  "id": 0,
}
```

Parameter content type:

application/json

```
"scheduleId": 0  
}
```

Click to set as parameter value

Try it out!

POST /occurrences

Create a new instance of the model and persist it into the data source.

Response Class (Status 200)

Model | Model Schema

```
{  
  "timestamp": "2016-07-12",  
  "status": 0,  
  "id": 0,  
  "scheduleId": 0  
}
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	<div style="border: 1px solid #ccc; width: 100px; height: 100px; margin: 0 auto;"></div> <p>Parameter content type: application/json</p>	Model instance data	body	Model   Model Schema <pre>{   "timestamp": "2016-07-12",   "status": 0,   "id": 0,   "scheduleId": 0 }</pre>

Click to set as parameter value

Try it out!

GET /occurrences/{id}

Find a model instance by id from the data source.

### Response Class (Status 200)

Model | Model Schema

```
{
  "timestamp": "2016-07-12",
  "status": 0,
  "id": 0,
  "scheduleId": 0
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string
filter	<input type="text"/>	Filter defining fields and include	query	string

Try it out!

HEAD /occurrences/{id}

Check whether a model instance exists in the data source.

## Response Class (Status 200)

Model | Model Schema

```
{  
  "exists": true  
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string

Try it out!

PUT `/occurrences/{id}`

Update attributes for a model instance and persist it into the data source.

## Response Class (Status 200)

Model | Model Schema

```
{  
  "timestamp": "2016-07-12",  
  "status": 0,  
  "id": 0,  
  "scheduleId": 0  
}
```

Response Content Type



## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>data</b>	<div style="border: 1px solid #ccc; width: 100px; height: 100px; margin-bottom: 5px;"></div> <p>Parameter content type: <input type="text" value="application/json"/></p>	An object of model property name/value pairs	body	Model   Model Schema <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"><pre>{   "timestamp": "2016-07-12",   "status": 0,   "id": 0,   "scheduleId": 0 }</pre></div> <p>Click to set as parameter value</p>
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

DELETE **/occurrences/{id}**

Delete a model instance by id from the data source.

### Response Class (Status 200)

Model | Model Schema

```
{}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string

GET /occurrences/{id}/exists

Check whether a model instance exists in the data source.

### Response Class (Status 200)

Model | Model Schema

```
{
  "exists": true
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string

Try it out!

GET /occurrences/{id}/schedule

Fetches belongsTo relation schedule.

### Response Class (Status 200)

Model | Model Schema

```
{
  "name": "string",
  "filter": {},
  "format": "string",
  "frequency": "string",
  "day": 0,
}
```

```
"hour": 0,
"enabled": true,
"timezone": "string",
"recipients": [
  {}
],
"id": 0,
"reportId": 0
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
refresh	<input type="text"/>		query	boolean
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

[Try it out!](#)

GET </occurrences/change-stream>

[Create a change stream.](#)

### Response Class (Status 200)

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
options	<input type="text"/>		query	string

[Try it out!](#)

POST /occurrences/change-stream

Create a change stream.

### Response Class (Status 200)

Response Content Type

#### Parameters

Parameter	Value	Description	Parameter Type	Data Type
options	<input type="text"/>		formData	string

Try it out!

GET /occurrences/count

Count instances of the model matched by where from the data source.

### Response Class (Status 200)

Model | Model Schema

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

Response Content Type

#### Parameters

Parameter	Value	Description	Parameter Type	Data Type
where	<input type="text"/>	Criteria to match model instances	query	string

Try it out!

GET /occurrences/findOne

Find first instance of the model matched by filter from the data source.

### Response Class (Status 200)

Model | Model Schema

```
{
  "timestamp": "2016-07-12",
  "status": 0,
  "id": 0,
  "scheduleId": 0
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>	Filter defining fields, where, include, order, offset, and limit	query	string

Try it out!

POST /occurrences/update

Update instances of the model matched by where from the data source.

### Response Class (Status 200)

Model | Model Schema

```
{}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
where	<input type="text"/>	Criteria to match model instances	query	string
data	<div style="border: 1px solid gray; width: 100px; height: 100px; margin-bottom: 5px;"></div> <p>Parameter content type:</p> <input type="text" value="application/json"/>	An object of model property name/value pairs	body	<div style="display: flex; border-bottom: 1px solid gray;"><span style="border-right: 1px solid gray; padding: 2px 5px;">Model</span><span style="padding: 2px 5px;">Model Schema</span></div> <div style="border: 1px solid gray; padding: 5px; margin-top: 5px;"><pre>{   "timestamp": "2016-07-12",   "status": 0,   "id": 0,   "scheduleId": 0 }</pre></div> <p>Click to set as parameter value</p>

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",
  "id": 0,
  "facetId": 0
}
]
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>	Filter defining fields, where, include, order, offset, and limit	query	string

Try it out!

PUT /reports

Update an existing model instance or insert a new one into the data source.

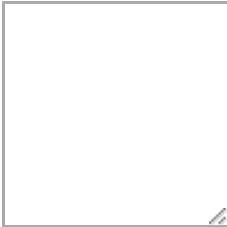
### Response Class (Status 200)

Model | Model Schema

```
{
  "title": "string",
  "urlSegmentName": "string",
  "filterDataSource": "string",
  "id": 0,
  "facetId": 0
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	 Parameter content type: <input type="text" value="application/json"/>	Model instance data	body	Model   Model Schema <pre>{   "title": "string",   "urlSegmentName": "string",   "filterDataSource": "string",   "id": 0,   "facetId": 0 }</pre> <p>Click to set as parameter value</p>

Try it out!

POST /reports

Create a new instance of the model and persist it into the data source.

### Response Class (Status 200)

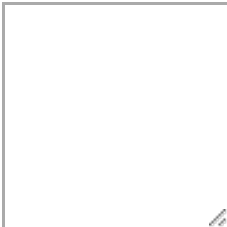
Model | Model Schema

```
{
  "title": "string",
  "urlSegmentName": "string",
  "filterDataSource": "string",
  "id": 0,
  "facetId": 0
}
```



Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	 Parameter content type: <input type="text" value="application/json"/>	Model instance data	body	Model   <b>Model Schema</b>

```
{
  "title": "string",
  "urlSegmentName": "string",
  "filterDataSource": "string",
  "id": 0,
  "facetId": 0
}
```

Click to set as parameter value

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",
  "id": 0,
  "facetId": 0
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string
filter	<input type="text"/>	Filter defining fields and include	query	string

Try it out!

HEAD /reports/{id}

Check whether a model instance exists in the data source.

### Response Class (Status 200)

Model | Model Schema

```
{
  "exists": true
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string

Try it out!

### Response Class (Status 200)

Model | Model Schema

```

{
  "title": "string",
  "urlSegmentName": "string",
  "filterDataSource": "string",
  "id": 0,
  "facetId": 0
}

```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	<div data-bbox="400 879 627 1107" data-label="Form"><input type="text"/></div> <p>Parameter content type: <input type="text" value="application/json"/></p>	An object of model property name/value pairs	body	Model   Model Schema <div data-bbox="1541 940 2024 1243" data-label="Code-Block"> <pre> {   "title": "string",   "urlSegmentName": "string",   "filterDataSource": "string",   "id": 0,   "facetId": 0 } </pre> </div> <p>Click to set as parameter value</p>
id	<input type="text" value="(required)"/>	PersistedModel id	path	string

Try it out!

DELETE /reports/{id}

Delete a model instance by id from the data source.

### Response Class (Status 200)

Model | Model Schema

```
{}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string

Try it out!

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

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>report id</b>	path	string
<b>format</b>	<input type="text" value="(required)"/>	<b>pdf or csv</b>	path	string
<b>state</b>	<input type="text" value="(required)"/>	<b>page state</b>	formData	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
------------------	--------	----------------	---------

204

Request was successful

Try it out!

GET /reports/{id}/exists

Check whether a model instance exists in the data source.

### Response Class (Status 200)

Model | Model Schema

```
{
  "exists": true
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string

Try it out!

GET /reports/{id}/facet

Fetches belongsTo relation facet.

### Response Class (Status 200)

Model | Model Schema

```
{
  "name": "string",
  "id": 0
}
```

```
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
refresh	<input type="text" value=""/>		query	boolean
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

Try it out!

GET /reports/{id}/facet/data

### Response Class (Status 200)

Model | Model Schema

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

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

<b>start</b>	<input type="text" value="(required)"/>	<b>Start Date</b>	query	string
<b>end</b>	<input type="text" value="(required)"/>	<b>End Date</b>	query	string
granularity	<input type="text"/>	Default value to all	query	string

Try it out!

GET /reports/{id}/schedules

Queries schedules of report.

### Response Class (Status 200)

Model | Model Schema

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

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>		query	string
id	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

Try it out!

POST /reports/{id}/schedules

Creates a new instance in schedules of this model.

### Response Class (Status 200)

Model | Model Schema

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



Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	<div style="border: 1px solid #ccc; width: 100px; height: 100px; margin-bottom: 5px;"></div> <p>Parameter content type:</p> <input type="text" value="application/json"/>		body	<div style="display: flex; border-bottom: 1px solid #ccc;"><div style="border-right: 1px solid #ccc; padding: 5px;">Model</div><div style="padding: 5px;">Model Schema</div></div> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 5px;"><pre>{   "name": "string",   "filter": {},   "format": "string",   "frequency": "string",   "day": 0,   "hour": 0,   "enabled": true,   "timezone": "string",   "recipients": [     {}   ],   "id": 0,   "reportId": 0 }</pre></div> <p style="font-size: small; text-align: right; margin-top: 5px;">Click to set as parameter value</p>

<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string
-----------	---	--------------------------	------	--------

Try it out!

DELETE **/reports/{id}/schedules** Deletes all schedules of this model.

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Request was successful		

Try it out!

GET /reports/{id}/schedules/{fk}

[Find a related item by id for schedules.](#)

### Response Class (Status 200)

Model | Model Schema

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

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>fk</b>	<input type="text" value="(required)"/>	<b>Foreign key for schedules</b>	path	string
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

Try it out!

PUT /reports/{id}/schedules/{fk}

Update a related item by id for schedules.

### Response Class (Status 200)

Model | Model Schema

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

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>fk</b>	<input type="text" value="(required)"/>	<b>Foreign key for schedules</b>	path	string
<b>data</b>	<div style="border: 1px solid #ccc; width: 100px; height: 100px; margin-bottom: 5px;"></div> Parameter content type: <input type="text" value="application/json"/>		body	Model   <b>Model Schema</b> <pre> {   "name": "string",   "filter": {},   "format": "string",   "frequency": "string",   "day": 0,   "hour": 0,   "enabled": true,   "timezone": "string",   "recipients": [     {}   ],   "id": 0,   "reportId": 0 } </pre> <p>Click to set as parameter value</p>
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

Try it out!

DELETE /reports/{id}/schedules/{fk}

Delete a related item by id for schedules.

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>fk</b>	<input type="text" value="(required)"/>	<b>Foreign key for schedules</b>	path	string

<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string
-----------	---	--------------------------	------	--------

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Request was successful		

[Try it out!](#)

GET /reports/{id}/schedules/count

Counts schedules of report.

### Response Class (Status 200)

Model | Model Schema

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

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
where	<input type="text"/>	Criteria to match model instances	query	string

<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string
-----------	---	--------------------------	------	--------

[Try it out!](#)

## Response Class (Status 200)

Model | Model Schema

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

Response Content Type 

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>		query	string
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

[Try it out!](#)

## Response Class (Status 200)

Model | Model Schema

```
{  
  "order": 0,  
  "title": "string",  
  "queryName": "string",  
  "component": "string",  
  "defaultParameters": {},  
  "layout": {},  
  "url": "string",  
  "id": 0,  
  "reportId": 0  
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	 Parameter content type: <input type="text" value="application/json"/>		body	Model   Model Schema <pre>{   "order": 0,   "title": "string",   "queryName": "string",   "component": "string",   "defaultParameters": {},   "layout": {},   "url": "string",   "id": 0,   "reportId": 0 }</pre>

Click to set as parameter value

<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string
-----------	---	--------------------------	------	--------

Try it out!

DELETE /reports/{id}/sections

Deletes all sections of this model.

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Request was successful		

Try it out!

GET /reports/{id}/sections/{fk}

Find a related item by id for sections.

### Response Class (Status 200)

Model | Model Schema

```
{
  "order": 0,
  "title": "string",
  "queryName": "string",
  "component": "string",
  "defaultParameters": {},
```



```
"layout": {},
"url": "string",
"id": 0,
"reportId": 0
}
```

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>fk</b>	<input type="text" value="(required)"/>	<b>Foreign key for sections</b>	path	string
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

[Try it out!](#)

PUT `/reports/{id}/sections/{fk}`

Update a related item by id for sections.

## Response Class (Status 200)

Model | Model Schema

```
{
  "order": 0,
  "title": "string",
  "queryName": "string",
  "component": "string",
  "defaultParameters": {},
  "layout": {},
  "url": "string",
  "id": 0,
  "reportId": 0
}
```

```
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>fk</b>	<input type="text" value="(required)"/>	<b>Foreign key for sections</b>	path	string
<b>data</b>	<div style="border: 1px solid gray; width: 100px; height: 100px; margin-bottom: 5px;"></div> <p>Parameter content type:</p> <input type="text" value="application/json"/>		body	<div style="display: flex; border-bottom: 1px solid gray;"><div style="border-right: 1px solid gray; padding-right: 5px;">Model</div><div>Model Schema</div></div> <div style="border: 1px solid gray; padding: 5px; margin-top: 5px;"><pre>{   "order": 0,   "title": "string",   "queryName": "string",   "component": "string",   "defaultParameters": {},   "layout": {},   "url": "string",   "id": 0,   "reportId": 0 }</pre></div> <p style="font-size: small; margin-top: 5px;">Click to set as parameter value</p>
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

Try it out!

DELETE `/reports/{id}/sections/{fk}`

Delete a related item by id for sections.

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------

<b>fk</b>	<input type="text" value="(required)"/>	Foreign key for sections	path	string
<b>id</b>	<input type="text" value="(required)"/>	PersistedModel id	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
204	Request was successful		

[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": "000000000000" }]}
```

**Response Class (Status 200)**

Model | Model Schema

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

Response Content Type

## Parameters

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

Try it out!

GET `/reports/{id}/sections/count`

Counts sections of report.

## Response Class (Status 200)

Model | Model Schema

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

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
where	<input type="text"/>	Criteria to match model instances	query	string
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

Try it out!

GET /reports/change-stream

[Create a change stream.](#)

## Response Class (Status 200)

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
options	<input type="text"/>		query	string

Try it out!

POST /reports/change-stream

Create a change stream.

### Response Class (Status 200)

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
options	<input type="text"/>		formData	string

Try it out!

GET /reports/count

Count instances of the model matched by where from the data source.

### Response Class (Status 200)

Model | Model Schema

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

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
where	<input type="text"/>	Criteria to match model instances	query	string

Try it out!

GET

[/reports/findOne](#)

Find first instance of the model matched by filter from the data source.

### Response Class (Status 200)

Model | Model Schema

```
{
  "title": "string",
  "urlSegmentName": "string",
  "filterDataSource": "string",
  "id": 0,
  "facetId": 0
}
```

Response Content Type 

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>	Filter defining fields, where, include, order, offset, and limit	query	string

[Try it out!](#)

POST

[/reports/update](#)

Update instances of the model matched by where from the data source.

### Response Class (Status 200)

Model | Model Schema

```
{}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
where	<input type="text"/>	Criteria to match model instances	query	string
data	<div style="border: 1px solid gray; width: 100px; height: 100px; margin-bottom: 5px;"></div> <p>Parameter content type:</p> <input type="text" value="application/json"/>	An object of model property name/value pairs	body	<div style="border: 1px solid gray; padding: 5px;"><p>Model   Model Schema</p><pre>{   "title": "string",   "urlSegmentName": "string",   "filterDataSource": "string",   "id": 0,   "facetId": 0 }</pre></div>

Click to set as parameter value

Try it out!

## schedule

GET /schedules

Find all instances of the model matched by filter from the data source.

Response Class (Status 200)

Model | Model Schema



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

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>	Filter defining fields, where, include, order, offset, and limit	query	string

Try it out!

PUT /schedules

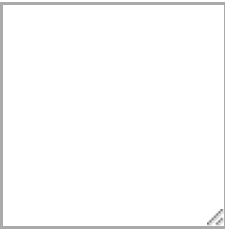
Update an existing model instance or insert a new one into the data source.

Response Class (Status 200)

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

Response Content Type

### Parameters

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

Parameter content type:

```
{
  "name": "string",
  "filter": {},
  "format": "string",
  "frequency": "string",
  "day": 0,
  "hour": 0,
  "enabled": true,
  "timezone": "string",
}
```

```
"recipients": [
  {}
],
"id": 0,
"reportId": 0
}
```

Click to set as parameter value

Try it out!

POST /schedules

Create a new instance of the model and persist it into the data source.

### Response Class (Status 200)

Model | Model Schema

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

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	<div style="border: 1px solid #ccc; width: 100px; height: 100px; margin-bottom: 5px;"></div> <p>Parameter content type:</p> <input type="text" value="application/json"/>	Model instance data	body	Model   Model Schema

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

Click to set as parameter value

[Try it out!](#)

GET /schedules/{id}

Find a model instance by id from the data source.

### Response Class (Status 200)

Model | Model Schema

```
{
  "name": "string",
  "filter": {},
  "format": "string",
```

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

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string
filter	<input type="text"/>	Filter defining fields and include	query	string

[Try it out!](#)

HEAD /schedules/{id}

Check whether a model instance exists in the data source.

## Response Class (Status 200)

Model | Model Schema

```
{
  "exists": true
}
```

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string

Try it out!

PUT /schedules/{id}

Update attributes for a model instance and persist it into the data source.

## Response Class (Status 200)

Model | Model Schema

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

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	<div style="border: 1px solid #ccc; width: 100px; height: 100px; margin-bottom: 5px;"></div> <p>Parameter content type:</p> <input type="text" value="application/json"/>	An object of model property name/value pairs	body	<span>Model</span>   <span>Model Schema</span> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"><pre>{   "name": "string",   "filter": {},   "format": "string",   "frequency": "string",   "day": 0,   "hour": 0,   "enabled": true,   "timezone": "string",   "recipients": [     {}   ],   "id": 0,   "reportId": 0 }</pre></div> <p style="font-size: small; text-align: right;">Click to set as parameter value</p>

<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string
-----------	---	--------------------------	------	--------

[Try it out!](#)

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

### Response Class (Status 200)

Model | Model Schema

```
{}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string

Try it out!

GET `/schedules/{id}/exists`

Check whether a model instance exists in the data source.

### Response Class (Status 200)

Model | Model Schema

```
{  
  "exists": true  
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string

Try it out!



GET /schedules/{id}/occurrences

Queries occurrences of schedule.

### Response Class (Status 200)

Model | Model Schema

```
[
  {
    "timestamp": "2016-07-12",
    "status": 0,
    "id": 0,
    "scheduleId": 0
  }
]
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>		query	string
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

Try it out!

POST /schedules/{id}/occurrences

Creates a new instance in occurrences of this model.

### Response Class (Status 200)

Model | Model Schema

```
{
```

```
"timestamp": "2016-07-12",
"status": 0,
"id": 0,
"scheduleId": 0
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	<div style="border: 1px solid gray; width: 100px; height: 100px; margin-bottom: 5px;"></div> <p>Parameter content type:</p> <input type="text" value="application/json"/>		body	Model   Model Schema <div style="border: 1px solid gray; padding: 5px; margin-top: 5px;"><pre>{   "timestamp": "2016-07-12",   "status": 0,   "id": 0,   "scheduleId": 0 }</pre></div> <p><small>Click to set as parameter value</small></p>
id	<input type="text" value="(required)"/>	PersistedModel id	path	string

[Try it out!](#)

DELETE /schedules/{id}/occurrences

Deletes all occurrences of this model.

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="(required)"/>	PersistedModel id	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Request was successful		

Try it out!

GET /schedules/{id}/occurrences/{fk}

Find a related item by id for occurrences.

### Response Class (Status 200)

Model | Model Schema

```
{
  "timestamp": "2016-07-12",
  "status": 0,
  "id": 0,
  "scheduleId": 0
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>fk</b>	<input type="text" value="(required)"/>	<b>Foreign key for occurrences</b>	path	string
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

Try it out!

PUT /schedules/{id}/occurrences/{fk}

Update a related item by id for occurrences.

## Response Class (Status 200)

Model | Model Schema

```
{  
  "timestamp": "2016-07-12",  
  "status": 0,  
  "id": 0,  
  "scheduleId": 0  
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>fk</b>	<input type="text" value="(required)"/>	<b>Foreign key for occurrences</b>	path	string
data	<div style="border: 1px solid gray; width: 100px; height: 100px; margin-bottom: 5px;"></div> <p>Parameter content type:</p> <input type="text" value="application/json"/>		body	Model   Model Schema <div style="border: 1px solid gray; padding: 5px; margin-top: 5px;"><pre>{   "timestamp": "2016-07-12",   "status": 0,   "id": 0,   "scheduleId": 0 }</pre></div> <p>Click to set as parameter value</p>
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

Try it out!

DELETE /schedules/{id}/occurrences/{fk}

Delete a related item by id for occurrences.

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>fk</b>	<input type="text" value="(required)"/>	<b>Foreign key for occurrences</b>	path	string
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Request was successful		

[Try it out!](#)

GET /schedules/{id}/occurrences/count

Counts occurrences of schedule.

### Response Class (Status 200)

Model | Model Schema

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

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
where	<input type="text"/>	Criteria to match model	query	string

instances

<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string
-----------	---	--------------------------	------	--------

Try it out!

GET /schedules/{id}/report

Fetches belongsTo relation report.

### Response Class (Status 200)

Model | Model Schema

```
{
  "title": "string",
  "urlSegmentName": "string",
  "filterDataSource": "string",
  "id": 0,
  "facetId": 0
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
refresh	<input type="text" value=""/>		query	boolean
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

Try it out!

GET /schedules/change-stream

[Create a change stream.](#)

### Response Class (Status 200)

Response Content Type

#### Parameters

Parameter	Value	Description	Parameter Type	Data Type
options	<input type="text"/>		query	string

[Try it out!](#)

POST /schedules/change-stream

[Create a change stream.](#)

### Response Class (Status 200)

Response Content Type

#### Parameters

Parameter	Value	Description	Parameter Type	Data Type
options	<input type="text"/>		formData	string

[Try it out!](#)

GET /schedules/count

[Count instances of the model matched by where from the data source.](#)

### Response Class (Status 200)

[Model](#) | [Model Schema](#)

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

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
where	<input type="text"/>	Criteria to match model instances	query	string

Try it out!

GET </schedules/findOne>

Find first instance of the model matched by filter from the data source.

## Response Class (Status 200)

Model | Model Schema

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



```
"reportId": 0
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>	Filter defining fields, where, include, order, offset, and limit	query	string

Try it out!

POST **/schedules/update**

Update instances of the model matched by where from the data source.

### Response Class (Status 200)

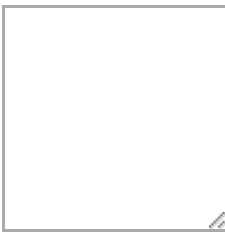
Model | Model Schema

```
{}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
where	<input type="text"/>	Criteria to match model instances	query	string
data		An object of model property name/value pairs	body	Model   Model Schema



Parameter content type:

application/json

Try it out!

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

Click to set as parameter value

## setting

GET /settings

Find all instances of the model matched by filter from the data source.

### Response Class (Status 200)

Model | Model Schema

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

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>	Filter defining fields, where, include, order, offset, and limit	query	string

Try it out!

PUT /settings

Update an existing model instance or insert a new one into the data source.

### Response Class (Status 200)

Model | Model Schema

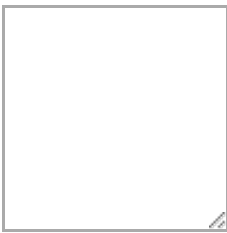
```
{  
  "key": "string",  
  "values": {}  
}
```

Response Content Type

### Parameters

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!

POST /settings

Create a new instance of the model and persist it 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
data		Model instance data	body	Model   Model Schema

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

Parameter content type:

application/json

Click to set as parameter value

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
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string
filter	<input type="text"/>	Filter defining fields and include	query	string

Try it out!

HEAD /settings/{id}

Check whether a model instance exists in the data source.

### Response Class (Status 200)

Model | Model Schema

```
{  
  "exists": true  
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string

Try it out!

PUT /settings/{id}

Update attributes for a model instance and persist it into the data source.

### Response Class (Status 200)

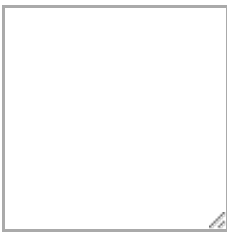
Model | Model Schema

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

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
data			body	



An object of model property name/value pairs

Model | Model Schema

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

Click to set as parameter value

Parameter content type:

application/json

**id**

(required)

**PersistedModel id**

path

string

Try it out!

DELETE /settings/{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
<b>id</b>	(required)	<b>Model id</b>	path	string

Try it out!

GET /settings/{id}/exists

Check whether a model instance exists in the data source.

## Response Class (Status 200)

Model | Model Schema

```
{  
  "exists": true  
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string

Try it out!

GET /settings/change-stream

[Create a change stream.](#)

## Response Class (Status 200)

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
options	<input type="text"/>		query	string

Try it out!

POST /settings/change-stream

[Create a change stream.](#)



## Response Class (Status 200)

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
options	<input type="text"/>		formData	string

Try it out!

GET /settings/count

Count instances of the model matched by where from the data source.

## Response Class (Status 200)

Model | Model Schema

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

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
where	<input type="text"/>	Criteria to match model instances	query	string

Try it out!

GET /settings/findOne

Find first instance of the model matched by filter from the data source.

### Response Class (Status 200)

Model | Model Schema

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

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>	Filter defining fields, where, include, order, offset, and limit	query	string

Try it out!

POST /settings/update

Update instances of the model matched by where from the data source.

### Response Class (Status 200)

Model | Model Schema

```
{}
```

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
where	<input type="text"/>	Criteria to match model instances	query	string
data	<div style="border: 1px solid #ccc; width: 100px; height: 100px; margin-bottom: 5px;"></div> <p>Parameter content type:</p> <input type="text" value="application/json"/>	An object of model property name/value pairs	body	<p>Model   Model Schema</p> <pre>{   "key": "string",   "values": {} }</pre> <p>Click to set as parameter value</p>

[Try it out!](#)

## system

GET	/systems	Find all instances of the model matched by filter from the data source.
<h3>Response Class (Status 200)</h3> <p>Model   Model Schema</p> <pre>[   {     "id": "string",     "type": "string",     "version": "string",     "location": "string",     "user": "string"   } ]</pre>		

```
}  
]
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>	Filter defining fields, where, include, order, offset, and limit	query	string

Try it out!

PUT **/systems**

Update an existing model instance or insert a new one into the data source.

### Response Class (Status 200)

Model | Model Schema

```
{  
  "id": "string",  
  "type": "string",  
  "version": "string",  
  "location": "string",  
  "user": "string"  
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter	Data Type
-----------	-------	-------------	-----------	-----------

Type

---

data Model instance data body Model Model Schema

Parameter content type:

```
{
  "id": "string",
  "type": "string",
  "version": "string",
  "location": "string",
  "user": "string"
}
```

Click to set as parameter value

[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

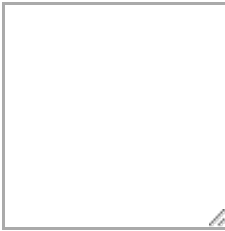
```
{
  "id": "string",
  "type": "string",
  "version": "string",
  "location": "string",
  "user": "string"
}
```

Response Content Type

Parameters

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------

data



Parameter content type:

application/json

Try it out!

Model instance data

body

Model | Model Schema

```
{
  "id": "string",
  "type": "string",
  "version": "string",
  "location": "string",
  "user": "string"
}
```

Click to set as parameter value

GET /systems/{id}

Find a model instance by id from the data source.

### Response Class (Status 200)

Model | Model Schema

```
{
  "id": "string",
  "type": "string",
  "version": "string",
  "location": "string",
  "user": "string"
}
```

Response Content Type application/json

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	(required)	<b>Model id</b>	path	string

filter

Filter defining fields and include

query

string

Try it out!

HEAD /systems/{id}

Check whether a model instance exists in the data source.

### Response Class (Status 200)

Model | Model Schema

```
{
  "exists": true
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	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",
"version": "string",
"location": "string",
"user": "string"
}
```

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	<div style="border: 1px solid #ccc; width: 100px; height: 100px; margin-bottom: 5px;"></div> <p>Parameter content type:</p> <input type="text" value="application/json"/>	An object of model property name/value pairs	body	Model   Model Schema
<b>id</b>	<input type="text" value="(required)"/>	<b>PersistedModel id</b>	path	string

```
{
  "id": "string",
  "type": "string",
  "version": "string",
  "location": "string",
  "user": "string"
}
```

Click to set as parameter value

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

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string

Try it out!

GET </systems/{id}/exists>

Check whether a model instance exists in the data source.

### Response Class (Status 200)

Model | Model Schema

```
{
  "exists": true
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string

Try it out!

GET </systems/change-stream>

Create a change stream.

### Response Class (Status 200)

Response Content Type

#### Parameters

Parameter	Value	Description	Parameter Type	Data Type
options	<input type="text"/>		query	string

Try it out!

POST /systems/change-stream

Create a change stream.

### Response Class (Status 200)

Response Content Type

#### Parameters

Parameter	Value	Description	Parameter Type	Data Type
options	<input type="text"/>		formData	string

Try it out!

GET /systems/count

Count instances of the model matched by where from the data source.

### Response Class (Status 200)

Model | Model Schema

```
{
```

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

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
where	<input type="text"/>	Criteria to match model instances	query	string

Try it out!

GET /systems/findOne

Find first instance of the model matched by filter from the data source.

### Response Class (Status 200)

Model | Model Schema

```
{
  "id": "string",
  "type": "string",
  "version": "string",
  "location": "string",
  "user": "string"
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter			query	

Filter defining fields, where, include, order, offset, and limit

string

Try it out!

POST

/systems/update

Update instances of the model matched by where from the data source.

### Response Class (Status 200)

Model | Model Schema

```
{}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
where	<input type="text"/>	Criteria to match model instances	query	string
data	<div data-bbox="414 1050 638 1278" data-label="Form"><input type="text"/></div> <p>Parameter content type: <input type="text" value="application/json"/></p>	An object of model property name/value pairs	body	<p>Model   Model Schema</p> <pre>{   "id": "string",   "type": "string",   "version": "string",   "location": "string",   "user": "string" }</pre>

Click to set as parameter value

Try it out!

## user

GET /users

Find all instances of the model matched by filter from the data source.

### Response Class (Status 200)

Model | Model Schema

```
[
  {
    "realm": "string",
    "username": "string",
    "credentials": {},
    "challenges": {},
    "email": "string",
    "emailVerified": true,
    "status": "string",
    "created": "2016-07-12",
    "lastUpdated": "2016-07-12",
    "id": 0
  }
]
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>	Filter defining fields, where, include, order, offset, and limit	query	string

Try it out!

PUT /users

Update an existing model instance or insert a new one into the data source.

## Response Class (Status 200)

Model | Model Schema

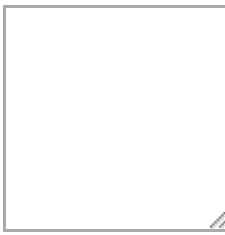
```
{
  "realm": "string",
  "username": "string",
  "credentials": {},
  "challenges": {},
  "email": "string",
  "emailVerified": true,
  "status": "string",
  "created": "2016-07-12",
  "lastUpdated": "2016-07-12",
  "id": 0
}
```

Response Content Type

## Parameters

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

```
{
  "realm": "string",
  "username": "string",
```



Parameter content type:

application/json

Try it out!

```
"credentials": {},
"challenges": {},
"email": "string",
"emailVerified": true,
"status": "string",
"created": "2016-07-12",
"lastUpdated": "2016-07-12",
"id": 0
}
```

Click to set as parameter value

POST /users

Create a new instance of the model and persist it into the data source.

Response Class (Status 200)

Model | Model Schema

```
{
  "realm": "string",
  "username": "string",
  "credentials": {},
  "challenges": {},
  "email": "string",
  "emailVerified": true,
  "status": "string",
  "created": "2016-07-12",
  "lastUpdated": "2016-07-12",
  "id": 0
}
```

Response Content Type application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	<div style="border: 1px solid #ccc; width: 100px; height: 100px; margin-bottom: 5px;"></div> <p>Parameter content type:</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div>	Model instance data	body	Model   Model Schema

```
{
  "realm": "string",
  "username": "string",
  "credentials": {},
  "challenges": {},
  "email": "string",
  "emailVerified": true,
  "status": "string",
  "created": "2016-07-12",
  "lastUpdated": "2016-07-12",
  "id": 0
}
```

[Click to set as parameter value](#)

[Try it out!](#)

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

### Response Class (Status 200)

Model | Model Schema

```
{
  "realm": "string",
  "username": "string",
  "credentials": {},
  "challenges": {},
  "email": "string",
  "emailVerified": true,
```



```
"status": "string",
"created": "2016-07-12",
"lastUpdated": "2016-07-12",
"id": 0
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<input type="text" value="(required)"/>	<b>Model id</b>	path	string
filter	<input type="text"/>	Filter defining fields and include	query	string

[Try it out!](#)

HEAD /users/{id}

Check whether a model instance exists in the data source.

### Response Class (Status 200)

Model | Model Schema

```
{
  "exists": true
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------

id

(required)

Model id

path

string

Try it out!

PUT /users/{id}

Update attributes for a model instance and persist it into the data source.

### Response Class (Status 200)

Model | Model Schema

```

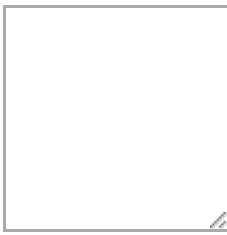
{
  "realm": "string",
  "username": "string",
  "credentials": {},
  "challenges": {},
  "email": "string",
  "emailVerified": true,
  "status": "string",
  "created": "2016-07-12",
  "lastUpdated": "2016-07-12",
  "id": 0
}

```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
data		An object of model property name/value pairs	body	Model   Model Schema <pre> {   "realm": "string", </pre>



Parameter content type:

application/json

```
"username": "string",
"credentials": {},
"challenges": {},
"email": "string",
"emailVerified": true,
"status": "string",
"created": "2016-07-12",
"lastUpdated": "2016-07-12",
"id": 0
}
```

Click to set as parameter value

<b>id</b>	(required)	<b>User id</b>	path	string
-----------	------------	----------------	------	--------

Try it out!

DELETE /users/{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
<b>id</b>	(required)	<b>Model id</b>	path	string

Try it out!

GET /users/{id}/accessTokens

Queries accessTokens of user.

### Response Class (Status 200)

Model | Model Schema

```
[
  {
    "id": "string",
    "ttl": 1209600,
    "created": "2016-07-12",
    "userId": 0
  }
]
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>		query	string
<b>id</b>	<input type="text" value="(required)"/>	<b>User id</b>	path	string

Try it out!

POST /users/{id}/accessTokens

Creates a new instance in accessTokens of this model.

### Response Class (Status 200)

Model | Model Schema

```
{
```

```
"id": "string",
"ttl": 1209600,
"created": "2016-07-12",
"userId": 0
}
```

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	<div style="border: 1px solid gray; width: 100px; height: 100px; margin-bottom: 5px;"></div> <p>Parameter content type:</p> <input type="text" value="application/json"/>		body	Model   Model Schema
id	<input type="text" value="(required)"/>	User id	path	string

```
{
  "id": "string",
  "ttl": 1209600,
  "created": "2016-07-12",
  "userId": 0
}
```

Click to set as parameter value

Try it out!

DELETE /users/{id}/accessTokens

Deletes all accessTokens of this model.

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="(required)"/>	User id	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Request was successful		

Try it out!

GET /users/{id}/accessTokens/{fk}

[Find a related item by id for accessTokens.](#)

### Response Class (Status 200)

Model | Model Schema

```
{
  "id": "string",
  "ttl": 1209600,
  "created": "2016-07-12",
  "userId": 0
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>fk</b>	<input type="text" value="(required)"/>	<b>Foreign key for accessTokens</b>	path	string
<b>id</b>	<input type="text" value="(required)"/>	<b>User id</b>	path	string

Try it out!

### Response Class (Status 200)

Model | Model Schema

```

{
  "id": "string",
  "ttl": 1209600,
  "created": "2016-07-12",
  "userId": 0
}

```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>fk</b>	<input type="text" value="(required)"/>	<b>Foreign key for accessTokens</b>	path	string
<b>data</b>	<div data-bbox="412 965 636 1193" data-label="Image"></div> <p>Parameter content type: <input type="text" value="application/json"/></p>		body	Model   Model Schema <div data-bbox="1579 1026 2027 1294" data-label="Code-Block"> <pre> {   "id": "string",   "ttl": 1209600,   "created": "2016-07-12",   "userId": 0 } </pre> </div> <p>Click to set as parameter value</p>
<b>id</b>	<input type="text" value="(required)"/>	<b>User id</b>	path	string

Try it out!

DELETE /users/{id}/accessTokens/{fk}

Delete a related item by id for accessTokens.

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>fk</b>	<input type="text" value="(required)"/>	<b>Foreign key for accessTokens</b>	path	string
<b>id</b>	<input type="text" value="(required)"/>	<b>User id</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Request was successful		

[Try it out!](#)

GET /users/{id}/accessTokens/count

Counts accessTokens of user.

### Response Class (Status 200)

Model | Model Schema

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

Response Content Type

### Parameters



Parameter	Value	Description	Parameter Type	Data Type
where	<input type="text"/>	Criteria to match model instances	query	string
<b>id</b>	<b>(required)</b>	<b>User id</b>	path	string

Try it out!

GET /users/{id}/exists

Check whether a model instance exists in the data source.

### Response Class (Status 200)

Model | Model Schema

```
{
  "exists": true
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<b>(required)</b>	<b>Model id</b>	path	string

Try it out!

GET /users/change-stream

Create a change stream.

### Response Class (Status 200)

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
options	<input type="text"/>		query	string

Try it out!

POST /users/change-stream

Create a change stream.

### Response Class (Status 200)

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
options	<input type="text"/>		formData	string

Try it out!

GET /users/confirm

Confirm a user registration with email verification token.

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>uid</b>	<input type="text" value="(required)"/>		query	string
<b>token</b>	<input type="text" value="(required)"/>		query	string
redirect	<input type="text"/>		query	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
------------------	--------	----------------	---------

204 Request was successful

Try it out!

GET /users/count

Count instances of the model matched by where from the data source.

## Response Class (Status 200)

Model | Model Schema

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

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
where	<input type="text"/>	Criteria to match model instances	query	string

Try it out!

GET /users/findOne

Find first instance of the model matched by filter from the data source.

## Response Class (Status 200)

Model | Model Schema

```
{
  "realm": "string",
  "username": "string",
  "credentials": {},
  "challenges": {},
  "email": "string",
  "emailVerified": true,
  "status": "string",
  "created": "2016-07-12",
  "lastUpdated": "2016-07-12",
  "id": 0
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>	Filter defining fields, where, include, order, offset, and limit	query	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

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>credentials</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">(required)</div> <p>Parameter content type: <input type="text" value="application/json"/></p>		body	<div style="display: flex; border-bottom: 1px solid #ccc;"><div style="border-right: 1px solid #ccc; padding: 2px 5px;">Model</div><div style="padding: 2px 5px;">Model Schema</div></div> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">{ }</div> <p>Click to set as parameter value</p>
include	<input type="text"/>	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!

POST </users/reset>

Reset password for a user with email.

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>options</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">(required)</div>		body	Model   Model Schema <div style="border: 1px solid #ccc; padding: 5px; display: inline-block;">{ }</div> <small>Click to set as parameter value</small>
	Parameter content type: <input type="text" value="application/json"/>			

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Request was successful		
<input type="button" value="Try it out!"/>			

POST **/users/update**

Update instances of the model matched by where from the data source.

### Response Class (Status 200)

Model | Model Schema

{ }

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------

where

Criteria to match model instances

query

string

data

An object of model property name/value pairs

body

Model | Model Schema

Parameter content type:

```

{
  "realm": "string",
  "username": "string",
  "credentials": {},
  "challenges": {},
  "email": "string",
  "emailVerified": true,
  "status": "string",
  "created": "2016-07-12",
  "lastUpdated": "2016-07-12",
  "id": 0
}

```

Click to set as parameter value

Try it out!

## zdXml

GET /zdXmIs

Response Class (Status 200)

Model | Model Schema

```

[
  {}
]

```

Response Content Type

Try it out!

POST /zdXmIs

### Response Class (Status 200)

Model | Model Schema

```
{}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
options	<div style="border: 1px solid #ccc; width: 100px; height: 100px; margin: 5px 0;"></div> <p>Parameter content type:</p> <input type="text" value="application/json"/>		body	Model   Model Schema <div style="border: 1px solid #ccc; width: 100px; height: 30px; margin: 5px 0; display: flex; align-items: center; justify-content: center;">{ }</div> <p>Click to set as parameter value</p>

Try it out!

GET /zdXmIs/{container}

### Response Class (Status 200)



Model | Model Schema

```
{}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
container	<input type="text"/>		path	string

Try it out!

DELETE /zdXmIs/{container}

### Response Class (Status 200)

Model | Model Schema

```
{  
  "undefined": {}  
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
container	<input type="text"/>		path	string

Try it out!

GET /zdXmls/{container}/download/{file}

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
container	<input type="text"/>		path	string
file	<input type="text"/>		path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Request was successful		

Try it out!

GET /zdXmls/{container}/files

### Response Class (Status 200)

Model | Model Schema

```
[  
  {}  
]
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
container	<input type="text"/>		path	string

Try it out!

GET /zdXmls/{container}/files/{file}

### Response Class (Status 200)

Model | Model Schema

```
{}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
container	<input type="text"/>		path	string
file	<input type="text"/>		path	string

Try it out!

DELETE /zdXmls/{container}/files/{file}

### Response Class (Status 200)

Model | Model Schema

```
{  
  "undefined": {}  
}
```

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
container	<input type="text"/>		path	string
file	<input type="text"/>		path	string

Try it out!

POST /zdXmls/{container}/upload

### Response Class (Status 200)

Model | Model Schema

```
{  
  "result": {}  
}
```

Response Content Type

Try it out!

GET /zdXmls/getAjaxRequest

### Response Class (Status 200)

Model | Model Schema

```
{}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>system_id</b>	<input type="text" value="(required)"/>	<b>System ID</b>	query	string

[Try it out!](#)

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

# Index

## A

access API [6](#)  
access token [7](#)  
API dialog box [9](#)  
API report parameters [14](#)

## C

common tasks [13](#)

## D

data endpoint [18](#)

## I

Identity [16](#)

## N

navigation [6](#)

## R

Report [16](#)  
reports [12](#)

## S

SCI [6](#)  
section ID [17](#)  
SmartCell Insight [6](#)

## U

user interface [6](#)



Copyright © 2017. Ruckus Wireless, Inc.  
350 West Java Drive, Sunnyvale, CA

[www.ruckuswireless.com](http://www.ruckuswireless.com)