

IAP/REST Fundamentals

This page describes certain fundamental issues about using the IAP/REST API. It includes the following sections:

- [What is REST?](#)
- [REST in IAP](#)
- [Making IAP/REST Calls](#)
- [IAP/REST Syntax](#)
- [An IAP Site](#)
- [Timestamps in IAP](#)

What is REST?

REST is an architectural style that allows applications to communicate effectively over HTTP by using HTTP methods (such as GET, PUT, and DELETE) to manage resources. REST is not a specification. However, it does have a number of constraints that together form the key guidelines for REST. The key constraints leveraged by IAP/REST are:

- **client-server model**, which requires that the responsibilities of the client application are completely separate from those of the server. All the client needs to know to access a service of the server is the resource's URI.
- **stateless**, which requires that the server does not store any information about the the last HTTP request made by the client. Each request made by the client must provide all of the information required to handle the request, either provided as part of the URI, query parameters, body, or headers. The server's response includes the information regarding the state, either included in the headers, response body, or status.
- **uniform interface**, which requires each resource to be identified using a single, unique URI, and all clients access all resources in the same manner.
- **layered system**, which allows you the option to use a layered system architecture, where the client cannot tell whether it is connected directly to the server or to an intermediary, such as a proxy or gateway.

REST components communicate by transferring a representation of a resource in a format matching one of an evolving set of standard data types, selected dynamically based on the capabilities or desires of the recipient and the nature of the resource.

REST in IAP

IAP/REST is an IoT access protocol based on REST. IAP/REST defines a standard set of resource types that are required for managing, monitoring, and controlling IoT devices. Resource types include devices, datapoints, alarms, logs, and schedules. Many IAP/REST resources represent a physical entity. For example, an IAP/REST device resource represents a physical IoT device. Some resources are contained within other resources. For example, an IAP/REST device resource may contain a number of IAP/REST datapoints, where each datapoint resource represents a datapoint on the physical IoT device.

Making IAP/REST Calls

Before being able to make any IAP/REST calls, you need to be logged into the CMS. See [Authentication \(Login/Logout\)](#) for more information.

IAP/REST Syntax

The syntax for IAP resources is as follows:

```
/iap / {Resource_Type} [ / {Resource_Detail} ] [ ? {Query_Parameters} ]
```

The syntax notation uses bold to indicate items that must be entered as shown, and terms in italics within parentheses to represent variables that must be replaced with an appropriate value. IAP is case-sensitive, so case must be as shown in the syntax descriptions.

Example:

```
GET /iap/floorplans/floors/{id}
```

Where:

iap	IAP URIs always begin with /iap , where iap is a constant keyword.
-----	---

<p><i>Resource_Type</i></p>	<p>IAP supports a variety of resource types such as devices, alarms, datapoints, etc. The supported resource types include those in the following list:</p> <table border="1" data-bbox="277 205 1164 1024"> <thead> <tr> <th>Resource Type</th> <th>Resource</th> </tr> </thead> <tbody> <tr><td>access</td><td>Geozone</td></tr> <tr><td>alarms</td><td>Alarms</td></tr> <tr><td>auth</td><td>Authentication</td></tr> <tr><td>customers</td><td>Customers</td></tr> <tr><td>devs</td><td>Devices, Device Capabilities, Datapoints</td></tr> <tr><td>diagnostics</td><td>Diagnostics</td></tr> <tr><td>floorplan</td><td>Floorplans</td></tr> <tr><td>generateKey</td><td>Customers</td></tr> <tr><td>gz</td><td>Geozone</td></tr> <tr><td>log</td><td>Log Destination, Log Events</td></tr> <tr><td>permission</td><td>Permissions</td></tr> <tr><td>reports</td><td>Reports</td></tr> <tr><td>role</td><td>Roles</td></tr> <tr><td>scevent</td><td>Schedule Events</td></tr> <tr><td>sch</td><td>Schedules</td></tr> <tr><td>user</td><td>Users</td></tr> <tr><td>version</td><td>Version</td></tr> </tbody> </table>	Resource Type	Resource	access	Geozone	alarms	Alarms	auth	Authentication	customers	Customers	devs	Devices, Device Capabilities, Datapoints	diagnostics	Diagnostics	floorplan	Floorplans	generateKey	Customers	gz	Geozone	log	Log Destination, Log Events	permission	Permissions	reports	Reports	role	Roles	scevent	Schedule Events	sch	Schedules	user	Users	version	Version
Resource Type	Resource																																				
access	Geozone																																				
alarms	Alarms																																				
auth	Authentication																																				
customers	Customers																																				
devs	Devices, Device Capabilities, Datapoints																																				
diagnostics	Diagnostics																																				
floorplan	Floorplans																																				
generateKey	Customers																																				
gz	Geozone																																				
log	Log Destination, Log Events																																				
permission	Permissions																																				
reports	Reports																																				
role	Roles																																				
scevent	Schedule Events																																				
sch	Schedules																																				
user	Users																																				
version	Version																																				
<p><i>Resource_Detail</i></p>	<p>Depending on the Resource Type, the <i>Resource Detail</i> may be zero or multiple levels deep, depending on the resource being specified. The following examples show the syntax for a user resource without any <i>Resource Detail</i>, and a device resource with two <i>Resource Detail</i> levels:</p> <div data-bbox="337 1203 1105 1409" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>For a user resource:</p> <pre>GET /iap/user</pre> <p>For a devs resource:</p> <pre>GET /iap/devs/floor/{id}</pre> </div> <p>Resource details can also include item reference fields. For example, the Datapoint Categories resource uses the following URI:</p> <div data-bbox="337 1476 1065 1520" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <pre>/iap/devs/{id}/ff/{block_name}/{block_index}/{datapoint_name}/cat</pre> </div> <p>The item reference fields in this URI are <i>id</i>, <i>block_name</i>, <i>block_index</i>, and <i>datapoint_name</i>. Item reference fields can contain path parameters as described in Queries and Parameters.</p>																																				
<p><i>Query_Parameters</i></p>	<p>Query parameters can be added to the end of the URI, following a '?' character. Multiple parameters can be specified by separating each parameter with an '&' character. For more information see Query Parameters.</p>																																				

An IAP Site

An IAP-based system consists of one or more sites. A site is typically a physical location, often providing shared Internet access to components within the site. Each site in an IAP-based system includes one or more segments, with one edge server controlling each segment.

Timestamps in IAP

All timestamps in IAP use the same format: YYYY-MM-DD HH:MM:SS.sss ZZZ, where YYYY is the four-digit year, MM is the two digit month, DD is the two-digit day of the month. HH is the hour of the day, always using the 24-hour clock, MM the number of minutes in the hour, SS the number of seconds in the minute, and sss the number of milliseconds within the second.

All numbers are padded with 0 to the left to meet the specified number of digits, if necessary. Number groups are separated with a single hyphen, colon, period, or a single space character.

ZZZ is the name of the timezone for wall time or 'UTC'. For information on configuring the timezone, see the **time** object in the [Segment Configuration](#) table.

Examples:

```
2016-09-15 16:33:18.321 BST  
2016-09-15 15:33:18.321 UTC
```

Most Recently Used (MRU) Timestamp

All messages published in the feedback channel must include a most recently used (mru) timestamp, using the standard IAP timestamp format. Messages posted in other channels may include an MRU timestamp. All MRU timestamps use UTC timestamps.

Most recipients of messages ignore the MRU, but some special tools might use the MRU, for example, to prune stale objects which may have resulted from the deletion of items. MRU timestamps are always ignored when assigned through the request channel.

Deletion of resources generally leaves one item behind, a status object which informs clients that reconnect in the future that the resource in question has been deleted. These items are typically pruned on a low frequency.