eleven

Sign in

Eleven > Developer Resources > Site Manager Resident Management API

Search

Articles in this section

Resident Management API - Overview

Follow

The Resident Management API allows Eleven partners and customers or third-party systems to manage resident data in Site Manager programmatically. Using the Resident management API eliminates the need for property staff to manually enter data in the Site Manager graphical user interface (GUI) and can be used with Integrated Mode in Site Manager.

If you want to start interacting with the Resident Management API immediately, navigate to the API Reference page and then download the Resident Management Postman Collection we have created.

Change Log

Date	Changes Made
11/11/2021	-Initial release

Base URL

All API requests must use the following base URL:

```
https://api.11os.com/v2
```

Request Body Objects

All request body objects must use the media type header `Content-Type: application/json`.

Responses

The Resident Management API returns JSON objects in all responses. For example, the `Get /team/{TeamId}/site/{SiteId}/resident/{ResidentId}` endpoint returns an individual resident account that looks like this:

Status: 200

Successful operation

JSON

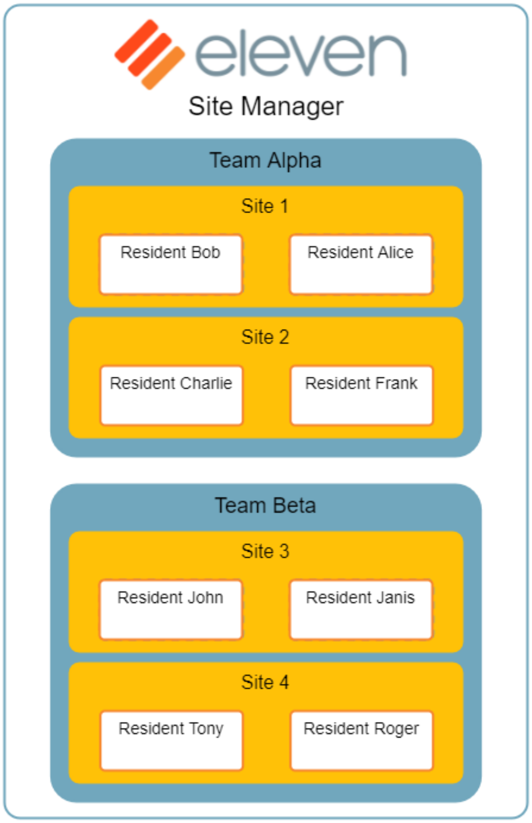
Pagination

Currently, the Resident Management API does not support pagination.

Organizational Schema

Site Manager is organized into a hierarchical structure of teams, sites, and residents.

Teams consist of **sites**, which then contain **residents**.



To access and manage resident data, you have to obtain the `TeamId` from the team and `SiteId` from the site where you want to manage residents. When managing individual residents, a `ResidentId` is also required.

PAN Mappings

Site Manager uses Personal Area Network (PAN) mappings to configure each resident's unit for internet access. PAN mappings link ElevenOS to the on-premises network hardware creating the configuration for each unit.

Like the Site Manager Organizational Schema, PAN mappings are organized into a hierarchical structure where **Buildings** consist of **Floors**, which then contain **Units**. The Building and Floor values are optional, and PAN Mapings must include Units - except for MDU SSO site types. PAN mapping configurations can have:

- multiple Buildings with Units and no Floors,
- multiple Floors with Units and no Buildings,
- multiple Units with no Buildings or Floors,
- or only Buildings without Floors or Units (MDU SSO).

It is also important to note that each Site Manager MDU site type may use PAN mappings differently and require different PAN mapping values. The differences between how each MDU site type uses PAN Mappings are based on the site's network topology, authorization service, or identity provider (IDP).

Network Topologies

There are two network topologies that MDU site types can use:

- **Centralized** - Centrally managed Wireless LAN Controller broadcasting a single site-wide SSID.
- **Distributed** - Individually managed gateway and access point (AP) with a unique SSID per unit or Building.

Authorization Services

MDU PPK-C Sites where the controller broadcasts a single site-wide SSID can use an external authorization service to manage passphrases per SSID/resident. The use of an authorization service allows you to generate multiple pre-shared keys for a single site-wide SSID. Some examples of authorization services are below:

- **Eleven Personal Pass Key (EKMS)** - Eleven's in-house authorization service.
- **Dynamic Pre-Shared Key (DPSK)** - Ruckus Cloudpath authorization service.
- **Easy Pre-Shared Key (EPSK)** - Cisco authorization service.

Identity Providers (IDP)

The MDU SSO site type uses an external IDP to authorize residents through Single-Sign-On (SSO).

PAN Mapping Values

Table-1 below shows each PAN Mapping value listed below is required unless otherwise noted:

Table-1 PAN Mapping Fields and Descriptions

PAN Mapping Field	Description
`Building`	(Optional for all MDU site types) Identifies the Floor to which a Unit belongs.
`Floor`	(Optional for all MDU site types) Identifies the Floor to which a Unit belongs.

`Unit`	Identifies the Unit allocated to a resident.
`VLAN`	Identifies the VLAN allocated to the unit.
`SSID`	Identifies the SSID associated with the unit.
`Passphrase`	The resident's passphrase. The PPK-C, PPK-D, and MDU SSO site types always set `Passphrase` to `null`, as they use external authorization services or an IDP to manage credentials.

Example Response

Below is an example response from the `/team/{{TeamId}}/site/{{SiteId}}/mappings` endpoint from a PPK-C site. In this example, there is one Building named Hudson, one Floor named 1, and multiple unique Units 100-103. Using any values outside of the ones returned in this response would result in an error.

Status: 200

Successful operation

JSON

Date and Time

You can schedule residents to onboard immediately or in the future. To onboard residents immediately, set the `ActivationDateUTC` to today's date. When you schedule a resident's onboarding date in the future, ElevenOS will trigger the onboarding process specified in the `ActivationDateUTC` value. You can choose to set the `DeactivationDateUTC` value to a specific date or leave it empty, indicating no date of service deactivation. Also, note that only one resident can occupy a unit on any given date or date range. So, if you choose to enter no Access End date for a unit, no residents can be allocated to that unit until you terminate and purge the current resident.

> **Info**
> For the remainder of this guide, it is essential to remember the terms used for the date a resident's service starts and ends may differ in some requests and responses. Below is a list of these terms:
>
> - `ActivationDate` or `Access Start` - Date that service activates in local site time.
> - `DeactivationDate` or `Access End` - Date that service activates in local site time.
> - `ActivationDateUTC` - Date that service activates in UTC.
> - `DeactivationDateUTC` - Date the service deactivates in UTC.

We recommend using the following default activation and deactivate times. These values can be adjusted to your liking, however.

- **Activation Time**: 6:00 am local time
- **Deactivation Time**: 11:59:59 pm local time

All times must use Coordinated Universal Time (UTC) formatting. Use the following UTC format for all time values:

```
YYYY-MM-DDThh:mm:ss.000Z
```

Example UTC:

```
2021-11-09T05:19:18.490939+00:00
```

Dates of Service Interactions

There are some interactions to remember about the activation and deactivation dates of service for a resident:

- The Activation Date cannot be blank and must contain a value.
- The Activation Date must come before the Deactivation Date.
- If the Deactivation Date is empty, the resident's service will never deactivate unless you manually terminate or enter a Deactivation Date.
- If you add a resident with Activation and Deactivation Dates in the past, Site Manager adds the resident with the specified Activation Date and removes the Deactivation Date.
- If you change the Activation Date from the current date or earlier to a future date, the resident's status changes from Current/Pending to Scheduled.

- If you change the Deactivation Date of a current resident to a past date, then the Deactivation Date becomes empty, and the resident's status remains unchanged. Therefore, the only way to terminate a resident is with the endpoints to Terminate or Terminate and Purge.

Errors

In the event errors occur, the below sections explain some error scenarios and what to do when encountering them.

HTTP Status Codes

The Resident Management API returns HTTP status codes that indicate the call was successful or an error occurred.

Code	Text	Description
200	Successful Operation	The request succeeded.
202	Accepted	The request was accepted but may not be processed yet.
400	Bad Request	The request was invalid. Check that all arguments, parameters, and JSON object values are correct.
403	Access Denied	Access is denied. The access token may be incorrect, or the level of permissions of the associated Site Manager account doesn't have sufficient privileges.
404	Not Found	The requested resource was not found.
406	Validation Error	The request failed due to a validation error such as an incorrect or misspelled item in the request body object payload.
409	Resource Conflict	The resident you are trying to add already exists at the site.
500	Internal Server Error	We had a problem processing your request. Please try again later or contact Eleven support.

Bulk Job Status Emails

The Resident Management API uses jobs for importing and performing bulk actions on multiple residents at once. The endpoints you use to import numerous residents and perform bulk actions on residents always send a job completion email to the account making API requests. This email details where and why any errors occur during a job.

Related Links

- Resident Management API - Quick Start Guide
- Resident Management API - Access and Permissions
- Resident Management API - Postman Collection

Was this article helpful?

✓ Yes ✕ No

0 out of 0 found this helpful

Have more questions? Submit a request

Return to top ↑

Recently viewed articles

Resident Management API - Quick Start Guide

Resident Management API - Access and Permissions

Role Templates for Single Sign-on (SSO)

Single Sign-on (SSO) for ElevenOS and Site Manager