RedSky Cloud Services Provisioning API

Programmer's Guide

Specification Version 2.0

August 1, 2021

Copyright © 2021 by RedSky Technologies, Inc.

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of RedSky Technologies, Inc., except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. For permission requests, write to the RedSky Technologies, Inc., addressed “Attention: Permissions Coordinator,” at the address below.

RedSky Technologies, Inc.
333 North Michigan Avenue, Suite 1600

Chicago, IL 60601

www.redskye911.com

Horizon Mobility®, E911 Anywhere®, MyE911® are registered trademarks of RedSky Technologies, Inc.

Revision History

Date	Version	Revision	Made By
09/16/19	1.0	• Initial Draft	Mike Koepke
08/01/21	2.0	• Removed API specifics to defer to online specifications • Added more concepts and know-how information • Address Fields definitions added	Mike Koepke

Table of Contents

1. Introduction

RedSky's Cloud Services provides 9-1-1 emergency calling solutions for Service Providers, with its Horizon Mobility® solution, and Enterprises, with E911 Anywhere®. This document explains how to access and use the RedSky's provisioning API to integrate your systems and applications its cloud services.

2. Scope

This guide is intended for use by Service Providers or Enterprises that would like to automate the provisioning of their data over an API as opposed to directly through the portal user interface. Through the API Sub-Tenants (Service Providers only), Users, Buildings and Locations can be added, edited, deleted, or viewed.

Throughout this document we use Horizon Mobility® to simplify the content of the document. Where we use Horizon Mobility® this also applies to E911 Anywhere®, unless specifically called out as being not applicable.

3. Online API Documentation

An online version of the API documentation is accessible at https://resources.e911cloud.com/apis/index.html. Here you will find the latest version of the endpoints, the parameters, and results.

When visiting the page, you will (as default) see documentation for the Authentication Service (./auth-service.json). This service is obviously key to all other API operations as you will need to authenticate and receive an access token.

Once authenticated you can utilize operations on the other services … to pull up the documentation for the other services, change the filename in the text bar at the top of the web page. The available services are:

- ./auth-service.json (Authentication service, displayed by default)
- ./admin-service.json (Administration service, for most organization level administration, configuration, and user administration)
- ./geography-service.json (Building and Location operations)
- ./mye911-service.json (Operations in support of mobile and desktop client non-corporate location management (personal locations) and location tracking)
- ./networking-service.json (Operations to configure mapping of locations to supported networking information (wifi, IP ranges, LLDP, mac address)

4. Technical Notes

4.1. JSON Data Format

All API endpoints and operations use the JSON format to pass or receive payload information. Client applications must be capable of generating and consuming data in this format. The endpoints do not accept XML-based content.

4.2. Secure Transport

Requests sent to Horizon Mobility® must be over HTTPS. Any requests over HTTP (not HTTPS) will be ignored and result in no response. Only connections using TLS version 1.2 or greater are supported.

4.3. REST Endpoint URLs

Clients sending requests to the RedSky Horizon Mobility® cloud should use the Server URL specific for their instance to make REST API calls. This Server URL may vary from RedSky Horizon Mobility® instance to instance. Upon establishing service with RedSky for Horizon Mobility®, the Server URL along with access credentials will be provided

5. Data Concepts

This section explains the data concepts used Horizon Mobility®. It is necessary to understand data relationships and concepts for proper provisioning.

5.1. Organizations

In Horizon Mobility® the top data element is an Organization. An Organization owns its Buildings, Locations, Users (Device Users), Alert Subscriptions, Alert Templates, etc. as provisioned in the system. Every organization in the system is assigned their own unique Organization Identifier. When provisioning data through endpoint calls, an Organization ID is passed as a parameter in generally every endpoint call when adding, editing, deleting, searching, or listing children data elements.

5.2. Organizational Types (Horizon Mobility only)

Horizon Mobility® supports sub-tenanting whereby an Organization can have children organizations underneath it. These children can be either a Reseller/Business partner that they themselves have children or an end Customer receiving the 9-1-1 service coverage. A Subtenant behaves exactly like an Organization, except the Subtenant is a child of an Organization. The parent Organization has access to all its subtenant's children (and their children) and the ability to provision data for them. A subtenant only has access to its own data. For example, Organization "Tom's Telco" can add Locations for its Sub-Tenant "Acme Manufacturing". An Organization can also create new subtenants directly through endpoints, provided it has been enabled to perform this capability.

When provisioning an organization in Horizon Mobility®, there is an Organization Type assigned to Organization. There are three types of Organizations: Service Provider, Reseller/Business Partners, and Customers. An Organization can only create Organization Types underneath of lesser privilege. A Service Provider can create Resellers or Customer types, a Reseller can only create Reseller types and a Customer cannot create any children. Only Reseller/Business Partner and Customer types can be provisioned over the API. It is highly recommended that only Customer types be created over the API.

5.3. Device Users Vs Users

In Horizon Mobility®, as well as in normal conversions, the concept of user has numerous and a also a generalized meaning. The term users can be those that access the Horizon Mobility® Portal, they can be clients that make API calls, or they can be ultimate end user whose location is maintained by Horizon Mobility® and places 9-1-1 calls.

In Horizon Mobility®, the data concept of a Device User means the ultimate end user whose location is maintained by Horizon Mobility® and places 9-1-1 calls. For those end users (lower case intended) running the RedSky MyE911® client, they will need to be provisioned in Horizon Mobility® using the Device User endpoints.

5.4. Building And Locations

In Horizon Mobility®, a Location is defined as an address (street address, city, state, zip) plus an additional information called Location Information. Location Information is synonymous with Address Line 2 in USPS addressing. This Location Information are things like Floor 2, Room 3, Suite 100, Apt. 10b, etc. Location

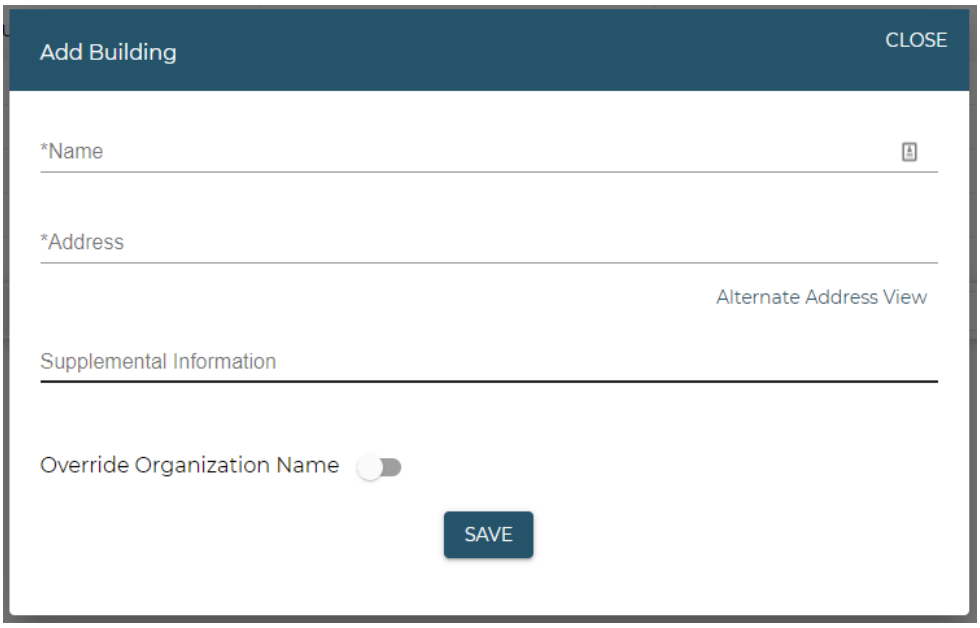
including Location Information is what is provided to a PSAP as the Dispatchable Location when an emergency call is made.

A Building in Horizon Mobility® is a logical grouping of Locations. Some organizations may wish to group their Locations together for ease of use when administering the Locations in the Horizon Mobility® Portal. When using Buildings and Locations together, the address can simply be associated with the Building and only the Location Information (Floor 2, Room 3, Suite 100, Apt. 10b, etc.) needs to be assigned to a Location. The Building endpoints are presented in this API guide, but this construct is not required, and the Location endpoints can be used on their own.

5.5. Address Provisioning

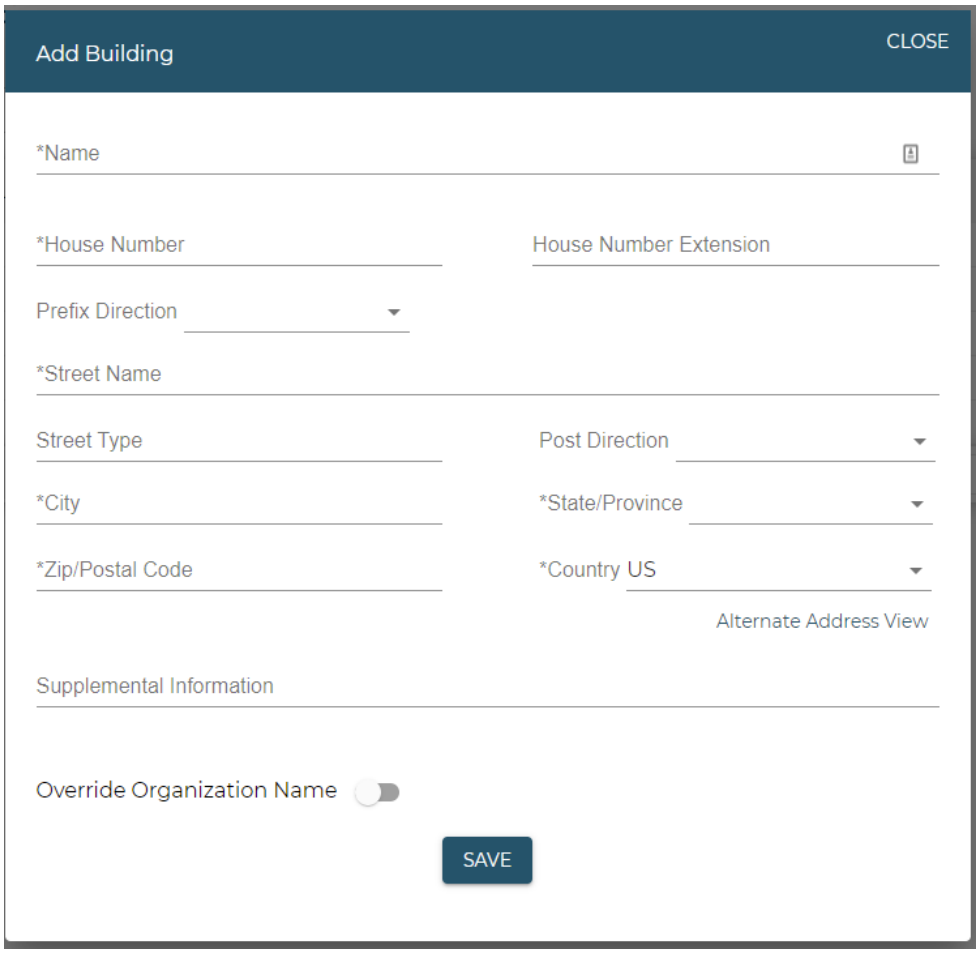
There are three mechanisms to submit the address when adding or editing buildings or locations. One form is a simplified, compact entry method where the street address, city, state/province, and zip/postal code are entered/submitted over the API. Both US and Canadian addresses are supported. It is not necessary to pass in the country code as it will be determined by the State/Province submitted.

For example, Address line is: 555 ½ W Main St, Anytown, IL 60001



Though the single line entry is generally successful, at times the parsing and decomposing of addresses does not adequately break the address into its correct components. The address can also be submitted with street, city, state, and zip using the compact format

Final method is supported where the address is submitted broken up in the parsed format.



Using 555 ½ W Main St, Anytown, IL 60001 from above:

- House Number – 555
- House Number Extension – ½
- Prefix Direction - W
- Street Name – Main
- Street Type – St
- Post Direction – blank

- City – Anytown
- State/Province – IL
- Zip/Postal Code – 60001
- Country - US

When an address is submitted, the address will be normalized and converted to a format valid for delivery to the PSAP. The API call will return the normalized address with the final address result along with any city, state, address or zip code corrections. Should the submitted address result in several possible options, the “ambiguous” flag will be set with the possible options returned. If the submitted address is completely unusable, the address list return will be empty. Future version of the APIs will return an address status as to the quality and state of the address submitted and results returned.

6. API Security

6.1. Authentication

All provisioning API calls will require the client application to present a security token when it makes the request. Horizon Mobility® Services uses JSON Web Tokens (JWT) to provide authorization for requests. Your application must pass the correct token in the HTTP header when making provisioning requests to the RedSky Services.

Your application can request a web token from the RedSky Authorization Service with your organizationID username and password credentials for a user that has been created in the UI.

To obtain credentials Look for the “login-controller” section of the page, and the ‘POST /login’ operation. When executed successfully this operation returns a json structure which will have the bearer token (a jwt is valid for 15 min) needed for the authorization header in other requests, as well as a “refresh” token – a special string that can be used (for 12 hours) to acquire a new bearer token without performing another “login” with credentials (see “token-controller”, POST /token/refresh).

Some data returned in the successful login response that you will need to use:

- accessToken (your jwt credential)

- refreshTokenInfo.id (your refresh token string)

- userProfileTO.company.id (identifier for your user’s organization in the system)

Example Request

```
{

  "password": "string",
  "username": "string"
}
```

Example Response

```
HTTP/1.1 200 OK

{

"status": "Success", "token_type":"bearer", "expires_in":"3600", "access_token":

"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI2NWQ4ZDBkYi1lYTAzLTQ2N2UtOWI3MS0xM2JkZ

jk5NmMxYmQiLCJhdWQiOiJhZG1pbi11aSIsInJzUm9sZSI6Ik9SR19BRE1JTiIsImlzcyI6ImNpcnJ1cy1jb25z
dW1lci1jcmVkZW50aWFsIiwicnNPcmciOiIwYjQzNWNhNy0xOWFkLTQ2MWUtODNmZS1hYWYyODI4ZWJlODAiLCJ
yc09yZ05hbWUiOiJDaXNjbyBTeXN0ZW1zIEluYy4iLCJleHAiOjE1NjI4NjY3MDYsImlhdCI6MTU2Mjg2NTgwNi wic-
nNTdWJqZWN0VHlwZSI6IlVTRVJfUFJPRklMRV9JRCIsInVzZXJuYW1lIjoidGVzdEBjaXNjby5jb20ifQ.6E
MwGF3ubyR5UrIeVENrFmVMqhY5_YWuz8YXXo1WgGo"
```

The response returns the following fields:

access_token - The access token issued by the authorization server.

token_type - The type of the token, which will always be “bearer” for Provisioning Calls.

expires_in - The lifetime in seconds of the access token. For example, the value "3600" denotes that the access token will expire in one hour from the time the response was generated.

5.1 Authorization

You must provide the token as part of the HTTP Header each time you make an API call.

```
Authorization: Bearer <access_token>
```

When your application calls the RedSky Horizon Mobility® Provisioning API, the API first validates that the JWT token is valid. If so, the API call will be executed. If not, the following error message will be returned:

```
HTTP/1.1 401 Unauthorized Content-Type: application/json

{

"message": "Unauthorized"
```

7. Address Field Definitions

Field	Size	Notes
House Number	10	House Number
House Number Suffix	4	House number extension (e.g. ½)
Prefix Directional	3	Leading street direction prefix. Valid entries: N S E W NE NW SE SW
Street Name	60	Valid Service address for the Calling Party Number
Street Type	4	Valid street abbreviation as defined by U. S. Postal Service Publication 28
Post Directional	3	Leading street direction prefix. Valid entries: N S E W NE NW SE SW
City	32	Valid service community name as identified by the U.S. Postal Service
State	2	U.S. State or Canadian province abbrevia-tion.
Zip	10	Postal or Zip Code
Country	2	Either US or CA
Location Description	60	Unstructured (free-format) information about a location - a textual address or additional "address" information.
Caller Name	32	Customer or Organization Name. Should be formatted as LastName, FirstName
Building	40	Name of the building the street address re-fers to

8. Organization Endpoints

RedSky Horizon Mobility® offers an extensive suite of API operations to allow you to perform various operations including add, edit, delete and view the list of Organizations and Subtenants.

Operations for managing organizations are accessed through the company-service using the company-controller. Due to some legacy concepts the use of company snuck into this controller. Generally, organizations are used throughout but the controller and endpoints reference the term company

The most common endpoints that will be used are:

- GET /company - Find organizations by search term, parent or type
- POST /company – Add an organization
- PUT /company – Edit an organization
- GET /company/{id} – Retrieve an organization by its organization id
- Delete /company/{id} – Delete an organization by its organization id

9. Device User Endpoints

RedSky Horizon Mobility® offers an extensive suite of API operations to allow you to perform various operations including add, edit, delete and view the list of device users.

Operations for managing users are accessed through the company-service using the device-user-controller.

The most common endpoints that will be used are:

- POST /deviceUser – Add a device user
- PUT /deviceUser – Edit a device user
- GET /deviceUser/{id} – Retrieve a device user by its id
- Delete /deviceUser/{id} – Delete an organization by its id
- GET /deviceUser/email/{email} – Retrieve a device user by its email address
- GET /deviceUser/phone/{phone} – Retrieve a device user by its phone number
- GET /deviceUser/altId/{altId} – Retrieve a device user by its alternative id

10. Building Endpoints

RedSky Horizon Mobility® offers an extensive suite of API operations to allow you to perform various operations including add, edit, delete and view the list of Buildings for an Organization.

Operations for managing users are accessed through the geography-service using the building-controller.

The most common endpoints that will be used are:

- POST /buildings – Add a building using single line address
- PUT /buildings – Edit a building using single line address
- GET /buildings/{id} – Retrieve a building by its id
- Delete /buildings/{id} – Delete a building by its id
- POST /buildings /compactAddress – Add a l building using compact address format
- PUT /buildings /compactAddress – Edit a building using compact address format
- POST /buildings /parsedAddress – Add a building using parsed address format
- PUT /buildings /parsedAddress – Edit a building using parsed address format

11. Location Endpoints

RedSky Horizon Mobility® offers an extensive suite of API operations to allow you to perform various operations including add, edit, delete and view the list of locations.

Operations for managing users are accessed through the geography-service using the location-controller.

The most common endpoints that will be used are:

- POST /locations – Add a location using single line address
- PUT /locations – Edit a location using single line address
- GET /locations/{id} – Retrieve a location by its id
- Delete /locations/{id} – Delete a location by its id
- GET /locations/tn/{tn} – Retrieve a location by its phone number
- GET /locations/search – Retrieve a location by a search term
- POST /locations/compactAddress – Add a location using compact address format
- PUT /locations/compactAddress – Edit a location using compact address format
- POST /locations/parsedAddress – Add a location using parsed address format
- PUT /locations/parsedAddress – Edit a location using parsed address format

12. Transport Response Codes

The Provisioning API Service runs on top of the HTTP/HTTPS protocol. Client applications submitting API requests must support the return and handling of standard HTTP protocol responses

12.1. 200 OK

If the format of the API request is valid and processed by Horizon Mobility®, a HTTP 200 OK response will be sent. When a 200 OK is returned, any results are expected back from the endpoint, such as in a GET, while bed contained in the response body.

12.2. 400 BAD REQUEST

If the server cannot or will not process the request due to something that is perceived to be a client error (e.g., malformed request syntax, invalid request message framing, or deceptive re-quest routing), a 400 Bad Request error is returned.

12.3. 401 UNAUTHORIZED

If the authorization token in the client request is missing or is not valid, a 401 Authorized error is returned.

12.4. 404 NOT FOUND

If the requested API endpoint is not valid or the path cannot be found, a 404 Not Found error is returned.

12.5. 500 ERROR

If an operational error occurs with Horizon Mobility® and it cannot process the API call at the time of the request, a 500 series error code will be returned.