



## INTRODUCTION

The Alianza REST API web services are a powerful set of tools that can be used to create and control all aspects of end-user accounts on Alianza's Cloud Voice Platform. REST is a simple way to interact between two systems. It is based on HTTP, and uses the URL structure and methods common on the Web. RESTful web services are generally preferable to SOAP and other web service standards as they are much simpler to understand and use.

Alianza's API allows service providers to securely, quickly and reliably integrate the Cloud Voice Platform with third-party back-office systems (e.g., CRM, billing systems databases, OSS, etc.). This integration automates workflow, removes error-prone steps, and enables scale—ultimately reducing the total cost of ownership of delivering and supporting VoIP services.

Alianza provides service provider customers with access to a development server and code samples for rapid prototyping as well as support from Alianza's REST experts. With these

tools, service providers can integrate with the Alianza platform in a matter of days or weeks.

The Alianza REST API contains over 20 resources. A resource is a collection of actions affecting similar objects. These resources set up accounts, submit and manage phone number port requests, and configure and manage a variety of options including call forwarding, soft phones, and E911 service. An overview of the key resources is in the table.

## END POINTS

An end point is a URL containing the resource and potentially one or more parameters used to focus the action taken on a specific object.

For example, the Account resource contains the following end points:

```
/v1/partition/{partitionId}/account/{id}
/v1/partition/{partitionId}/account/{id} accounthistorysearch
/v1/partition/{partitionId}/account/{id}/callingplanproduct/{callingPlanId}
/v1/partition/{partitionId}/account/{id}/status
/v1/partition/{partitionId}/account/search
```

Note: this is an example of the variety of end points, not an exhaustive list. See the full REST documentation for more.

## HEADER, BODY AND PARAMETERS

The URL alone does not always contain all the information necessary to complete an action. There is also information that is passed in the URL, or the header or the body of the HTTP message.

### Header

The header contains a definition of the content in the body as well as authentication credentials.

### Body

The body parameter passes the data of the object(s) being addressed. The dataset used in the body is defined by the model associated with each object. The complete model is viewable as the data type for the body parameter listed in each verb.

| RESOURCE                 | DESCRIPTION   |
|--------------------------|---|
| <b>Account</b>           | Account structure, new account activation   |
| <b>Calling Plan</b>      | Minutes and usage, blocked destinations   |
| <b>Device</b>            | Provisioning and device status  |
| <b>End User</b>          | Call forwarding, login credentials, voicemail, etc.   |
| <b>Hunt Groups</b>       | Create and manage Hunt Groups for business lines (requires Line Hardware and Line Appearance) |
| <b>IVR</b>               | Menus, hunt groups, schedules   |
| <b>Line Hardware</b>     | Add and manage endpoints (analog gateways or key systems) for business lines                  |
| <b>Line Appearance</b>   | Add and manage lines for business lines   |
| <b>Media</b>             | Upload and retrieve media for hold music and IVR Prompts                                      |
| <b>Report</b>            | Retrieve CSV formatted data   |
| <b>SIP Trunk</b>         | SIP Trunk concurrent call limits, SIP Trunk credentials                                       |
| <b>Telephone Number</b>  | LNP reactions, CSR updates  |
| <b>Voicemail Box</b>     | Add voicemail boxes and edit their settings   |
| <b>Voicemail Message</b> | Retrieve and manage voicemails  |

### **Model**

The model defines the acceptable data and format contained in the body of each request. The model typically contains all the applicable values associated with a particular object. The model may differ by HTTP method. For example, to list information about the account, using GET to the `/v1/partition/{partitionId}/account/{id}` without any information in the body. However, to make an update to the account, a PUT made to the same endpoint would require a JSON object in the body.

### **JSON**

JSON (JavaScript Object Notation) is a data syntax defining the titles and values of a dataset and is used by the Alianza API. See <http://www.json.org/> for details regarding syntax.

### **ENUM**

When a field requires data that is part of a list, or a limited set of data, the model definition will contain an enumeration, or list of the possible values.

### **Multipart Form**

The multipart form is used for file uploads. The response body contains an id referring to the file.

### **Path**

A path parameter modifies the URL to focus the action onto a specific set of data or an object.

The Alianza REST API predominately uses the ID of an object as a path parameter. IDs are used to reference a partition (belonging to an Alianza service provider customer) and an end-user account (residential or business customer).

For example, when referencing an account use the accountID, or ID of the account, to identify the account to be modified in the URL path. In the example `http://rest.alianza.com/v1/partition/{partitionId}/account/{id}/status` the `partitionId` is the unique identifier for the service provider customer and the `id` is the unique identifier of the account in question. This URL will return the status of a specific account for the specified service provider.

Example: If the partition ID value is 123 and the account ID is 456, the end point URL would be `http://rest.alianza.com/v1/partition/123/account/456/status`.

### **Query**

Query parameters are added to the URL to modify its usage. For example the "q" parameter is used with the `/v1/partition/{partitionId}/account/search` end point to search for accounts.

Example: To search for all accounts in the 123 partition containing the term "Smith" the endpoint URL would be: `http://rest.alianza.com/v1/partition/123/account/search?q=Smith`

Parameters available for each resource may differ by method. These parameters are listed in the parameters section inside the appropriate HTTP method for each resource.

## **HTTP METHODS**

The HTTP methods PATCH, GET, POST, PUT, and DELETE are used to perform actions on the objects of each resource.

### **GET**

An HTTP GET will retrieve the current state of an object as it exists in the database. No updates are made in the process as the data is just read and returned.

Parameters can be added to the end point to limit the results set. The request body is unused when using the HTTP GET. The response body contains the model of the object requested, which may contain a single object or a list of objects.

### **PATCH**

An HTTP PATCH will only update the model objects that were supplied and supports partial updates to models. Unlike PUT, which requires all fields, the PATCH method is used to update specific values in the model without affecting the other objects in the model.

### **POST**

When an HTTP POST is made to an end point, the object will be created.

The request body must contain the entire model in the object. Fields which are automatically set to a default settings, may be omitted. The response body contains the model of the object created or an error message if the POST was not successful.

### **PUT**

An HTTP PUT will replace the entire object with the information supplied in the PUT.

The request body must contain the entire object model, not just the values being updated. The response body contains the updated model of the object.

### **DELETE**

When an HTTP DELETE is made to an end point, the object is deleted. No other parameters are required.

## **MORE INFORMATION**

Alianza's documentation is dynamically created based on the current version of the Alianza API. This documentation can be found at <https://rest.alianza.com/rest/apidocs/>

For more information regarding REST, JSON, or questions regarding usage of the Alianza REST API contact your Alianza Client Services representative or Alianza support at 1 801-802-6400.