Aruba Central
Application Programming Interface
Copyright Information
© Copyright 2019 Hewlett Packard Enterprise Development LP.

Open Source Code
This product includes code licensed under the GNU General Public License, the GNU Lesser General Public License, and/or certain other open source licenses. A complete machine-readable copy of the source code corresponding to such code is available upon request. This offer is valid to anyone in receipt of this information and shall expire three years following the date of the final distribution of this product version by Hewlett Packard Enterprise Company. To obtain such source code, send a check or money order in the amount of US $10.00 to:

Hewlett Packard Enterprise Company
6280 America Center Drive
San Jose, CA 95002
USA
## Contents

**About this Guide** ................................................................. 5
- Contacting Support ................................................................. 5

**API Gateway** ................................................................. 6
- API Gateway and NB APIs ....................................................... 6
- List of Supported APIs ................................................................. 6
- Accessing API Gateway ................................................................. 8
  - Domain URLs ........................................................................ 8
  - Using OAuth 2.0 to Access API ................................................. 8
- Access and Refresh Tokens ................................................................. 8
- Creating an Application ................................................................. 9
- Obtaining Tokens ................................................................. 10
  - Offline Token Mechanism .......................................................... 10
  - Authorization Code Grant ......................................................... 10
- Refreshing a Token ................................................................. 12
  - Example .................................................................................. 13
- Deleting a Token ................................................................. 13
  - Example .................................................................................. 13
- Accessing APIs ................................................................. 14
  - Example .................................................................................. 14
- Viewing APIs ................................................................. 15
- Viewing Tokens ................................................................. 15
- Revoking Tokens ................................................................. 15
- Adding a New Token ................................................................. 16
- API Documentation ................................................................. 16
This guide describes how to use Aruba Central Application Programming Interface (API) to configure your applications.

**Contacting Support**

**Table 1: Contact Information**

<table>
<thead>
<tr>
<th>Category</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Site</td>
<td>arubanetworks.com</td>
</tr>
<tr>
<td>Support Site</td>
<td>support.arubanetworks.com</td>
</tr>
<tr>
<td>Airheads Social Forums and Knowledge Base</td>
<td>community.arubanetworks.com</td>
</tr>
<tr>
<td>North American Telephone</td>
<td>1-800-943-4526 (Toll Free) 1-408-754-1200</td>
</tr>
<tr>
<td>International Telephone</td>
<td>arubanetworks.com/support-services/contact-support/</td>
</tr>
<tr>
<td>Software Licensing Site</td>
<td>hpe.com/networking/support</td>
</tr>
<tr>
<td>End-of-life Information</td>
<td>arubanetworks.com/support-services/end-of-life/</td>
</tr>
<tr>
<td>Security Incident Response Team</td>
<td>Site: arubanetworks.com/support-services/security-bulletins/ Email: <a href="mailto:aruba-sirt@hpe.com">aruba-sirt@hpe.com</a></td>
</tr>
</tbody>
</table>
Aruba Central supports a robust set of REST APIs to enable users to build custom applications and integrate the APIs with their applications. The Aruba Central API framework uses OAuth protocol to authenticate and authorize third-party applications, and allows them to obtain secure and limited access to an Aruba Central service.

**API Gateway and NB APIs**

The **API Gateway** feature in Aruba Central supports the REST API for all Aruba Central services. This feature allows Aruba Central users to write custom applications, embed, or integrate the APIs with their own applications. The REST APIs support HTTP GET and POST operations by providing a specific URL for each query. The output for these operations is returned in the JSON format.

For secure access to the APIs, the Aruba Central API Framework plug-in supports OAuth protocol for authentication and authorization. The access tokens provide a temporary and secure access to the APIs. The access tokens have a limited lifetime for security reasons and the applications should use the refresh API to obtain new tokens periodically (every 2 hours).

**List of Supported APIs**

Aruba Central supports the following APIs for the managed devices.

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitoring</td>
<td>Gets network, client, and event details. It also allows you to manage labels and switches.</td>
</tr>
</tbody>
</table>
| Configuration | Allows you to configure and retrieve the following:  
|            |   - Groups  
|            |   - Templates  
|            |   - Devices |
| AppRF      | Gets Top N AppRF statistics.                                                  |
| Guest      | Gets visitor and session details of the portal.                              |
| MSP        | Allows you to manage and retrieve the following:  
|            |   - Customers  
|            |   - Users  
|            |   - Resources  
|            |   - Devices  
|            | Aruba has enforced a request limit for the following APIs:  
|            |   - GET /msp_api/v1/customers  
|            |   - GET /msp_api/v1/customers/{customer_id}/devices  
|            |   - GET /msp_api/v1/devices  
|            |   - PUT /msp_api/v1/customers/{customer_id}/devices  
|            | The maximum limit is set to 50 per API call. If you exceed this limit, the API call returns the HTTP error code 400 and the following error message: **LIMIT_REQUEST_EXCEEDED**. |
### Table 2: APIs and Description

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Management</td>
<td>Allows you to manage users and also allows you to configure various types of users with a specific level of access control.</td>
</tr>
<tr>
<td>Audit Event Logs</td>
<td>Gets a list of audit events and the details of an audit event.</td>
</tr>
<tr>
<td>Device Inventory</td>
<td>Gets device details and device statistics.</td>
</tr>
<tr>
<td>Licensing</td>
<td>Allows you to manage and retrieve subscription keys.</td>
</tr>
<tr>
<td>Presence Analytics</td>
<td>Allows you to configure the Presence Analytics application. It also retrieves site and loyalty data.</td>
</tr>
<tr>
<td>Device Management</td>
<td>Allows you to manage devices.</td>
</tr>
<tr>
<td>Firmware</td>
<td>Allows you to manage firmware.</td>
</tr>
<tr>
<td>Troubleshooting</td>
<td>Gets a list of troubleshooting commands for a specific type of device.</td>
</tr>
<tr>
<td>Notification</td>
<td>Gets notification alerts generated for events pertaining to device provisioning, configuration, and user management.</td>
</tr>
<tr>
<td>Clarity</td>
<td>Allows you to retrieve the status of the wireless connection for the devices added in Aruba Central and troubleshoot issues detected in the network.</td>
</tr>
<tr>
<td>Unified Communications</td>
<td>Retrieves data for all sessions for a specific period of time. It also retrieves the total number of clients who made calls in the given time range and gets the Lync/Skype for Business URL for the Aruba Central cluster that you are using.</td>
</tr>
<tr>
<td>Refresh API Token</td>
<td>Allows you to refresh the API token.</td>
</tr>
<tr>
<td>Reporting</td>
<td>Gets the list of configured reports for the given customer ID.</td>
</tr>
<tr>
<td>WAN Health</td>
<td>Allows you to the following:</td>
</tr>
<tr>
<td></td>
<td>■ Get list of configured WAN health policies.</td>
</tr>
<tr>
<td></td>
<td>■ Create a new WAN health policy.</td>
</tr>
<tr>
<td></td>
<td>■ Delete an existing WAN health policy.</td>
</tr>
<tr>
<td></td>
<td>■ Get the details of any specific WAN health policy.</td>
</tr>
<tr>
<td></td>
<td>■ Update an existing WAN health policy.</td>
</tr>
<tr>
<td></td>
<td>■ Delete recurring WAN health check policies on all the sites where it was configured.</td>
</tr>
<tr>
<td></td>
<td>■ Get policy schedule details.</td>
</tr>
<tr>
<td></td>
<td>■ Create a schedule for a WAN health policy.</td>
</tr>
<tr>
<td></td>
<td>■ Get statistics for WAN health cookie generated for a site.</td>
</tr>
<tr>
<td></td>
<td>■ Get WAN health test results.</td>
</tr>
<tr>
<td></td>
<td>■ Get WAN health test results for a specific site.</td>
</tr>
<tr>
<td>Network Health</td>
<td>Allows you to get data for all the labels and sites.</td>
</tr>
<tr>
<td>Webhook</td>
<td>Allows you to add, or delete Webhooks, and get or refresh Webhook tokens. See Webhooks on page 16 for further details on Webhook.</td>
</tr>
<tr>
<td>VisualRF</td>
<td>Allows you retrieve information on floor plans, location of APs, clients and rogue devices.</td>
</tr>
<tr>
<td>DPS Monitoring</td>
<td>Gets DPS compliance and session statistics for all the links of a device belonging to a specific policy.</td>
</tr>
</tbody>
</table>
Accessing API Gateway

To access API Gateway:

1. From the app selector, click Maintenance.
2. Click the API Gateway menu option. The API Gateway page is opens. The API Gateway page allows you to get new tokens and refresh the old tokens. To obtain a new token application, you must set authentication parameters for a user session.

The admin user profile of MSP has System Apps & Tokens tab which displays all the apps and tokens generated locally in the admin user profile. This tab also displays all the apps created in the non-admin user profiles. Clicking these apps lists out all the associated tokens created for the non-admin user profile.

For enabling API Gateway license, contact your Aruba sales representative.

Domain URLs

Table 3 shows the region-specific domain URLs for accessing API Gateway:

<table>
<thead>
<tr>
<th>Region</th>
<th>Domain Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>US-West-A (US-1 cluster zone)</td>
<td>app1-apigw.central.arubanetworks.com</td>
</tr>
<tr>
<td>US-West-B (US-2 cluster zone)</td>
<td>apigw-prod2.central.arubanetworks.com</td>
</tr>
<tr>
<td>Europe</td>
<td>eu-apigw.central.arubanetworks.com</td>
</tr>
<tr>
<td>APAC</td>
<td>api-ap.central.arubanetworks.com</td>
</tr>
<tr>
<td>China</td>
<td>apigw.central.arubanetworks.com.cn</td>
</tr>
<tr>
<td>Canada</td>
<td>apigw-ca.central.arubanetworks.com</td>
</tr>
</tbody>
</table>

The procedures described in this article use app1-apigw.central.arubanetworks.com as an example. Ensure that you use the appropriate domain URL when accessing API Gateway or generating tokens.

Using OAuth 2.0 to Access API

For secure access to the APIs, the Aruba Central API Framework plug-in supports OAuth protocol for authentication and authorization. OAuth 2.0 is a simple and secure authorization framework. It allows applications to acquire an access token for Aruba Central through a variety of work flows supported within the OAuth2 specification.

All OAuth2 requests must use the SSL endpoint available at https://app1-apigw.central.arubanetworks.com.

Access and Refresh Tokens

The access token is a string that identifies a user, app, or web page and is used by the app to access an API. The access tokens provide a temporary and secure access to the APIs.
The access tokens have a limited lifetime. If the application uses web server or user-agent OAuth authentication flows, a refresh token is provided during authorization that can be used to get a new access token.

If you are writing a long running applications (web app) or native mobile application you should refresh the token periodically. For more information, see Refreshing a Token on page 12.

Creating an Application

To create an application, complete the following steps:

1. From the app selector, click Maintenance.
2. Click the API Gateway menu option. The API Gateway page is displayed.
3. Click My Apps & Tokens.

The admin user will be able to create new apps for all the non-admin user by clicking + Add Apps & Tokens in the System Apps & Tokens tab.

4. Click + Add Apps & Tokens.
5. Enter a name for the application and the redirect URI in the Application Name and Redirect URI fields, respectively, and click Generate. A new application is created and added to the My Apps & Tokens table. The My Apps & Tokens table displays the following:

- **Name**—Name of the application. In non-admin user profile, the Application Name field contains the logged-in user name and is non-editable. Any new tokens generated in non-admin user profile will be associated with the same application name.
- **Client ID**—Unique ID for each application.
- **Client Secret**—Unique secret ID for each application.
- **Tokens**—An application can have multiple tokens. Click View Tokens to view the list of tokens.
- **Created At**—Date on which the application was created.
- **Actions**—Click the Delete icon on the row corresponding to an application and click Yes to delete that application.

After an application is created, go to the My Apps & Tokens page and click View Tokens. In the Token List pop-up window, you can view the user ID(s) associated to the application in the User Name column. An application can be associated to multiple users.

Only admin users will be able to generate tokens with multiple application names. In non-admin user profile, the Application Name field contains the user name and is non-editable. Any new tokens generated in non-admin user profile will be associated with the same application name. However, all the multiple application names and the associated tokens in non-admin user profiles from the earlier versions will still be retained in the Token List pop-up window.

After an application is created for a non-admin user in the admin user mode, go to the System Apps & Tokens page and click View Tokens. In the Token List pop-up window, you can view the user IDs associated to the application in the User Name column. An application can be associated to multiple users.
Obtaining Tokens

The users can generate the OAuth token using one of the following methods:

- Offline token download
- Authorization code grant

Offline Token Mechanism

To obtain tokens using the offline token method, complete the following steps:

1. From the app selector, click Maintenance.
2. Click the API Gateway menu option. The API Gateway page is displayed.
3. Click My Apps & Tokens.

In the MSP mode, the admin user profile has System Apps & Tokens tab which displays all the apps and tokens generated in all the non-admin user profiles in addition to the apps and tokens created in the admin user profile.

4. Click + Add Apps & Tokens.
5. Click View Tokens to view the tokens associated to a specific application. The Token List pop-up window opens and displays the following:
   - Token ID—Token ID of the application.
   - User Name—Name of the user to whom this token is associated to. An application can be associated to multiple users.
   - Generated At—Date on which the token was generated.
   - Revoke Token—Click Revoke Token and click Yes to revoke the token associated to a particular user. For example, if two users are associated to an application and if you want to remove access to a particular user, revoke the token associated to that user.
   - Download Token—Click Download Token to download the token.

Authorization Code Grant

The following section describes the procedure for obtaining the access token and refresh token using the authorization code grant mechanism:

**Step 1: Create an Application**

Create an application in Aruba Central. For more information, see Creating an Application on page 9.

Make a note of the following:

- Client ID
- Client Secret

**Step 2: Identify the Base URL**

The following example shows the base URL:

https://app1-apigw.central.arubanetworks.com
Step 3: Generating Client Credentials
To generate client credentials, use the following URI and the request method:

Request Method—POST  

POST Request Body(JSON):
{
  "customer_id": "<tenant_id>"
}

RequestHeader: (Values from login API request)
Set-Cookie: csrftoken=xxxx;session=xxxx;

Response Body(JSON):
{
  "client_id": "<new-client-id>",
  "client_secret": "<new-client-secret>"
}

Step 4: Log in and Authenticate Using Your Aruba Central Credentials
Append the base URL with the following:
/oauth2/authorize/central/api/login

URL:https://app1-apigw.central.arubanetworks.com/oauth2/authorize/central/api/login

This endpoint is accessible over SSL. The HTTP (non-SSL) connections are redirected to the SSL port. The endpoint validates the user session. For Aruba Central, the SSO authentication page is presented. After successful authentication, a consent page is shown requesting the resource owner (the customer who has logged in) to give access to the APIs.

Request: Craft an HTTP POST request containing the username, password, and client ID of the application.

Response: The response contains the Cross-Site Request Forgery (CSRF) token and session key that are necessary for obtaining the authorization code.

Step 5: Obtain an Authentication Code
Append the base URL with the following:
/oauth2/authorize/central/api

URL:https://app1-apigw.central.arubanetworks.com/oauth2/authorize/central/api

This endpoint is accessible over SSL. The HTTP (non-SSL) connections are redirected to the SSL port. This endpoint is a POST call to get an authorization code.

Request: Craft an HTTP POST request containing the customer ID, client ID, CSRF token, and session key.

Response: The response contains the authorization code that is necessary for obtaining the access token and refresh token.

Step 6: Exchange the Authentication Code for the Access and Refresh Tokens
Append the base URL with the following:
/oauth2/token

URL:https://app1-apigw.central.arubanetworks.com/oauth2/token
This endpoint is accessible over SSL. The HTTP (non-SSL) connections are redirected to the SSL port. The endpoint is a POST call to get the access token and refresh token using the authorization code obtained from the server. This exchange must be done within 300 seconds of obtaining the authorization code from step 4. Otherwise, the API returns an error.

**Request:** Craft an HTTP POST request containing the client ID, client secret, and authentication code.

**Response:** The response contains the access token and refresh token. The access token will expire in 2 hours.

### Refreshing a Token

To refresh the access token, access the following URL:

https://app1-apigw.central.arubanetworks.com/oauth2/token

This endpoint is accessible over SSL. The HTTP (non-SSL) connections are redirected to SSL port.

**Table 4: Refresh Tokens**

<table>
<thead>
<tr>
<th>URL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://app1-apigw.central.arubanetworks.com/oauth2/token">https://app1-apigw.central.arubanetworks.com/oauth2/token</a></td>
<td>The endpoint is a POST call to refresh the access token using the refresh token obtained from the step 2.</td>
</tr>
</tbody>
</table>

The query parameters for the API are as follows:

**Table 5: Query Parameters for Refresh Tokens**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>client_id</td>
<td>A unique hexadecimal string</td>
<td>A unique identifier that identifies the caller. The application developers can request a client ID and client secret key by registering with the Aruba Technical Support.</td>
</tr>
<tr>
<td>client_secret</td>
<td>A unique hexadecimal string</td>
<td>The client secret is a unique identifier provided to each developer at the time of registration. The application developers can request a client ID and client secret by registering with the Aruba Technical Support.</td>
</tr>
<tr>
<td>grant_type</td>
<td>refresh_token</td>
<td>The grant_type must be refresh_token to refresh the token.</td>
</tr>
<tr>
<td>refresh_token</td>
<td>refresh_token received from step 2</td>
<td>A string representing the authorization granted to the client by the resource owner.</td>
</tr>
</tbody>
</table>

A JSON dictionary with the following values is returned as a response.
### Parameter values Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>token_type</td>
<td>bearer</td>
<td>Identifies the token type. Only the bearer token type is supported. For more information, see <a href="https://tools.ietf.org/html/rfc6750">https://tools.ietf.org/html/rfc6750</a>.</td>
</tr>
<tr>
<td>refresh_token</td>
<td>string</td>
<td>Refresh tokens are credentials used to renew or refresh the access_token when it expires without going through the complete authorization flow. A refresh token is a string representing the authorization granted to the client by the resource owner.</td>
</tr>
<tr>
<td>expires_in</td>
<td>seconds</td>
<td>The expiration duration of the access tokens in seconds.</td>
</tr>
<tr>
<td>access_token</td>
<td>string</td>
<td>Access tokens are credentials used to access the protected resources. An access token is a string representing an authorization issued to the client.</td>
</tr>
</tbody>
</table>

### Example

**Method: POST**

```plaintext
https://app1-apigw.central.arubanetworks.com/oauth2/token?client_id=98273576d558401581c425d5bd9df213&grant_type=refresh_token&refresh_token=1272ddc5f4c94683b7ac3080f39503f9&client_secret=e20f3fad10dc4c41bf291a49e85a3b29
```

**Response**

```json
{
  "access_token": "xyz",
  "customer_id": "bearer",
  "access_token": "889479cac74e4b299723cc9a6f8f9d08",
  "expires_in": 7200
}
```

### Deleting a Token

To delete the access token, access the following URL:

**https://app1-apigw.central.arubanetworks.com/oauth2/token**

This endpoint is accessible over SSL. The HTTP (non-SSL) connections are redirected to SSL port. Customer ID is a string.

**Example**

**Method: DELETE**

**URL:** [https://app1-apigw.central.arubanetworks.com/oauth2/api/tokens](https://app1-apigw.central.arubanetworks.com/oauth2/api/tokens)

**JSON Body:**

```json
{
  "access_token": "<access_token_to_be_deleted>",
  "customer_id": "<customer_id_to_whom_token_belongs_to>
```

**Headers:**

Content-Type: application/json

X-CSRF-Token: <CSRF_token_obatained_from_login_API>

Cookie: "session=<session_obatained_from_login_API>"
Accessing APIs

To access the API, use the following URL:


This endpoint is accessible over SSL and the HTTP (non-SSL) connections are redirected to the SSL port.

<table>
<thead>
<tr>
<th>Table 6: Accessing the API</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>URL</strong></td>
</tr>
<tr>
<td><a href="https://app1-apigw.central.arubanetworks.com/">https://app1-apigw.central.arubanetworks.com/</a></td>
</tr>
</tbody>
</table>

The query parameters for the API are as follows:

<table>
<thead>
<tr>
<th>Table 7: Query Parameters for the API</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td>request_path</td>
</tr>
<tr>
<td>access_token</td>
</tr>
</tbody>
</table>

Example

Request: (Method=Get)

https://app1-apigw.central.arubanetworks.com/monitoring/v1/aps?access_token=e325c0fb3f1547b5b735de3221690c2f

Response:

```json
{
  "aps": [ 
    {
      "firmware_version": "6.4.4.4-4.2.3.1_54637",
      "group_name": "00TestVRK",
      "ip_address": "10.29.18.195",
      "labels": [ 
        "Filter_242",
        "Ziaomof",
        "roster",
        "242455",
        "Diegso"
      ],
      "macaddr": "6c:f3:7f:c3:5d:92",
      "model": "AP-134",
      "name": "6c:f3:7f:c3:5d:92",
      "radios": [ 
        {
          "band": 0,
          "index": 1,
          "macaddr": "6c:f3:7f:b5:d9:20",
          "status": "Down"
        }
      ]
    }
  ]
}```
"band": 1,
"index": 0,
"macaddr": "6c:f3:7f:b5:d9:30",
"status": "Down"
}
],
"serial": "AX0140586",
"status": "Down",
"swarm_id": "e3bf1ba201a6f85f4b5eaedead5e502d85a9ae58d8e1d8a0",
"swarm_master": true
},
"count": 1
}

Viewing APIs

To view the APIs managed through Aruba Central, complete the following steps:

1. From the app selector, click Maintenance.
2. Click the API Gateway menu option. The API Gateway page with the list of published APIs is displayed.
3. To view the details of an API, click Details.
4. To view the API documentation, click the link in the Documentation column next to the specific published API name. The documentation is displayed in a new window.

Viewing Tokens

To view tokens, complete the following steps:

1. From the app selector, click Maintenance.
2. Click the API Gateway menu option. The API Gateway page is displayed.
3. Click My Apps & Tokens.
4. To view tokens, click the View Tokens link for the specific App & Token name. The Token List pop-up window opens.

In MSP, the admin user profile has System Apps & Tokens tab which displays all the apps and tokens generated in all the non-admin user profiles in addition to the apps and tokens created in the admin user profile. To view all the tokens of admin and non-admin user, go to Maintenance > API Gateway > System Apps & Tokens.

Revoking Tokens

To revoke tokens, complete the following steps:

1. From the app selector, click Maintenance.
2. Click the API Gateway menu option. The API Gateway page is displayed.
3. Click My Apps & Tokens.
4. To view tokens, click the View Tokens link for the specific App & Token name. The Token List pop-up window opens.
5. To revoke tokens, click Revoke Token in the Token List window.
Adding a New Token

To add a new token, complete the following steps:

1. From the app selector, click Maintenance.
2. Click the API Gateway menu option. The API Gateway page is displayed.
3. Click My Apps & Tokens.

The admin user will be able to create new tokens for all the non-admin user by clicking + Add Apps & Tokens in the System Apps & Tokens tab.

4. Click + Add Apps & Tokens to add a new token.
5. Enter the application name in the Application Name box and then click Generate.

If you have registered a custom URI when creating a new app under System Apps and Tokens, the Redirect URI option is disabled for you in the My Apps and Tokens tab > Add Apps and Tokens > New Token. In such cases, the Redirect URI option in Add Apps and Tokens > New Token under My Apps and Tokens populates your already registered URI.

Only admin users can generate tokens with multiple application names. In a non-admin user profile, the Application Name field contains the username and is non-editable. Any new tokens generated in the non-admin user profile will be associated with the same application name.

After an application is created for a non-admin user, go to the System Apps & Tokens page and click View Tokens. In the Token List pop-up window, you can view the user IDs associated to the application in the User Name column.

API Documentation

For a complete list of APIs and the corresponding documentation, see https://app1-apigw.central.arubanetworks.com/swagger/central.

Webhooks

Webhooks allow you to implement event reactions by providing real-time information or notifications to other applications. Aruba Central allows you to create Webhooks and select Webhooks as the notification delivery option for all alerts.

Using Aruba Central, you can integrate Webhooks with other third-party applications such as ServiceNow, Zapier, IFTTT, and so on.

You can access the Webhooks service either through the Aruba Central UI or API Gateway. Aruba Central supports creating up to 10 Webhooks. To enable redundancy, Aruba Central allows you to add up to three URLs per Webhook.

From Aruba Central, you can add, list, or delete Webhooks; get or refresh Webhooks token; get or update Webhooks settings for a specific item; and test Webhooks notification.
Use the **Webhook** check box in the **Monitoring & Reports > Alerts > Configure Alerts** page to select Webhooks as the notification delivery option.

The following figure illustrates how Aruba Central integrates with third-party applications using Webhooks.

**Figure 1  Webhooks Integration**

This section includes the following topics:
- [Creating and Updating Webhooks Through the UI on page 17](#)
  - [Refreshing Webhooks Token Through the UI on page 18](#)
- [Creating and Updating Webhooks Through the API Gateway on page 18](#)
  - [List of Webhooks APIs on page 18](#)

**Creating and Updating Webhooks Through the UI**

To access the Webhooks service from the UI:
1. Go to **Maintenance > API Gateway**.
2. In the **Webhook** tab, click **+Webhook**.
   a. **Webhook Name**—Enter a name for the Webhook
   b. **URLs**—Enter the URL. Click + to enter another URL. You can add up to three URLs.
3. Click **Save**. The Webhooks is created and listed in the **Webhook** table.

The **Webhook** table displays the following information and also allows you to edit or delete Webhooks:
- **Name**—Name of the Webhooks.
- **Number of URL Entries**—Number of URLs in Webhooks. Click the number to view the list of URLs.
- **Updated At**—Date and time at which Webhooks was updated.
- **Webhook ID**—Webhooks ID.
- **Token**—Webhooks token. Webhooks token enables header authentication and the third-party receiving service must validate the token to ensure authenticity.
- **Edit**—Click the Edit icon in the last column to edit the Webhook. You can refresh the token and add URLs. Click Save to save the changes.
- **Delete**—Click the Delete icon in the last column and click Yes to delete the Webhook.

### Refreshing Webhooks Token Through the UI

To refresh Webhooks token through the UI:

1. Go to **Maintenance > API Gateway > Webhook** tab.
2. In the **Webhook** table, click the edit icon in the **Token** column.
3. In the pop-up window, click the refresh icon next to the token. The token is refreshed.

### Creating and Updating Webhooks Through the API Gateway

The following HTTP methods are defined for Aruba Central API Webhooks resource:

- **GET**
- **POST**
- **PUT**
- **DELETE**

You can perform CRUD operation on the Webhooks URL configuration. The key configuration elements that are required to use API Webhooks service are Webhooks URL and a shared secret.

A shared secret token is generated for the Webhooks URL when you register for Webhooks. A hash key is generated using SHA256 algorithm by using the payload and the shared secret token. The API required to refresh the shared secret token is provided for a specific Webhooks configuration. You can choose the frequency at which you want to refresh the secret token.

To access and use the API Webhooks service:

1. Go to **Maintenance > API Gateway**.
2. In the **APIs** tab, click the **Swagger** link under the **Documentation** header. The **Swagger** website opens.
3. In the **Swagger** website, select **Webhook** from the **URL** drop-down list. All available Webhooks APIs are listed under **API Reference**.

For more information on Webhooks APIs, refer to [https://app1-apigw.central.arubanetworks.com/swagger/central](https://app1-apigw.central.arubanetworks.com/swagger/central).

### List of Webhooks APIs

Aruba Central supports the following Webhooks APIs:

- **GET /central/v1/webhooks**—Gets a list of Webhooks.

  The following is a sample response:

  ```json
  {
    "count": 1,
    "settings": [
      {
        "wid": "e26450be-4dac-435b-ac01-15d8f9667eb8",
        "name": "AAA",
        "updated_ts": 1523956927,
        "urls": [  
          
        ]
      }
  ]
  ```

---

**Note:**

For more information on Webhooks APIs, refer to [https://app1-apigw.central.arubanetworks.com/swagger/central](https://app1-apigw.central.arubanetworks.com/swagger/central).
POST /central/v1/webhooks—Creates Webhooks.
The following is a sample response:
{
    "name": "AAA",
    "wid": "e829a0f6-1e36-42fe-bafd-631443cbd581"
}

DELETE /central/v1/webhooks/{wid}—Deletes Webhooks.
The following is a sample response:
{
    "wid": "e26450be-4dac-435b-ac01-15d8f9667eb8"
}

GET /central/v1/webhooks/{wid}—Gets Webhooks settings for a specific item.
The following is a sample response:
{
    "wid": "e26450be-4dac-435b-ac01-15d8f9667eb8",
    "name": "AAA",
    "updated_ts": 1523956927,
    "urls": [
        "https://example.org/webhook1",
        "https://example.org/webhook1"
    ],
    "secure_token": "KEu5ZPTi44U04MnMi0qz"
}

PUT /central/v1/webhooks/{wid}—Updates Webhooks settings for a specific item.
The following is a sample response:
{
    "name": "AAA",
    "wid": "e829a0f6-1e36-42fe-bafd-631443cbd581"
}

GET /central/v1/webhooks/{wid}/token—Gets the Webhooks token for the Webhooks ID.
The following is a sample response:
{
    "name": "AAA",
    "secure_token": "{"token": "zf8jz8jzf8j", "ts": 1523957233}" }
- **GET /central/v1/webhooks/{wid}/ping**—Tests the Webhooks notification and returns whether success or failure.

  The following is a sample response:

  "Ping Response [{url: 'https://example.org', status: 404}]"

---

**Sample Webhooks Format for a New Alert Generation**

**URL POST <webhook-url>**

**Custom Headers**

Content-Type: application/json
X-Central-Service: Alerts
X-Central-Event: Radio-Channel-Utilization
X-Central-Delivery-ID: 72d3162e-cc78-11e3-81ab-4c9367dc0958
X-Central-Delivery-Timestamp: 2016-07-12T13:14:19-07:00
X-Central-Customer-ID: <#####>

**Body**

```json
{
  "id": "AV6fTbgEG8ncx_zH2-g-",
  "type": "RADIO_NOISE_FLOOR",
  "setting_id": "12800236-1253",
  "device_id": "NPAE000001",
  "timestamp": 1505914259,
  "operation": "create",
  "state": "open",
  "severity": "critical",
  "extra": {
    "_band": "5 GHz",
    "duration": "5",
    "name": "NPAE000001",
  }
}```