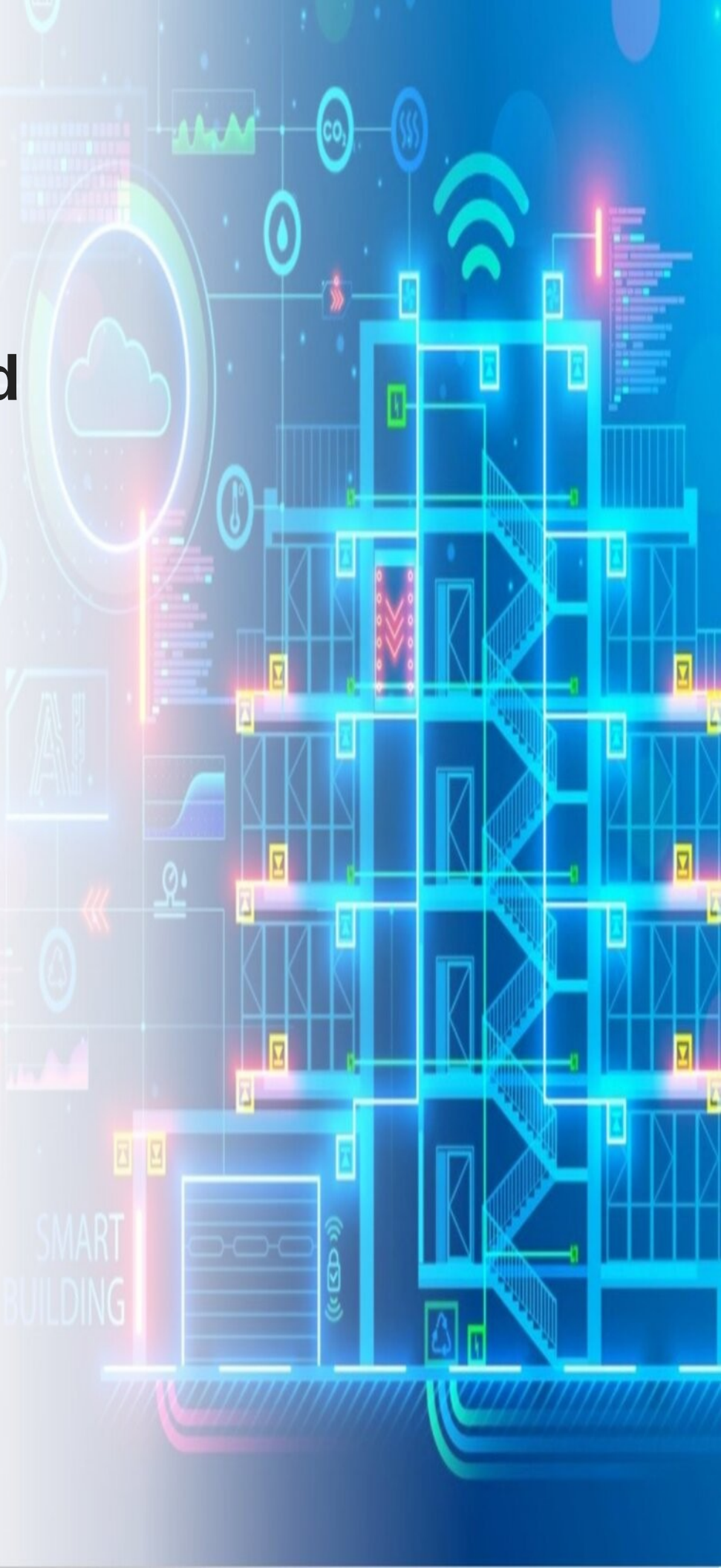
SHASTA CLOUD

INTEGRATION GUIDE

Shasta Cloud API How-To Guide

Creating API tokens, navigating Integration Docs, and querying the Telemetry Service in Shasta Cloud



DOCUMENT TYPE	API Integration Guide
PLATFORM	Shasta Cloud (Staging)
LAST UPDATED	April 23, 2026
VERSION	1.0

Contents

Table of Contents

Overview

Shasta Cloud provides a comprehensive set of RESTful APIs that let administrators and developers programmatically access network telemetry, manage infrastructure, query connected clients, retrieve event logs, and more. APIs are documented with the **OpenAPI Specification (OAS 3.0)** and are accessible through the **Integration Docs** section of the platform. Authentication is handled via Bearer tokens issued from **Admin > Integrations**.

Base URL: api-stg.shastacloud.com **Auth: Bearer Token** **Spec: OAS 3.0** **Version: 1.1.4**

Step 1 — Creating an API Token

Before you can make any API calls, you must create an API token. Tokens are managed in the Admin > Integrations section of the Shasta Cloud platform and must be included as a Bearer token on the Authorization header of every API call.

1.1 Navigate to Admin > Integrations

In the left navigation menu, click Admin to expand it, then click Integration. This opens the Integrations page with the API tab selected by default. The table shows each existing token's Name, Description, Created Date, Expiry Date, and Status (Enabled or Expired).

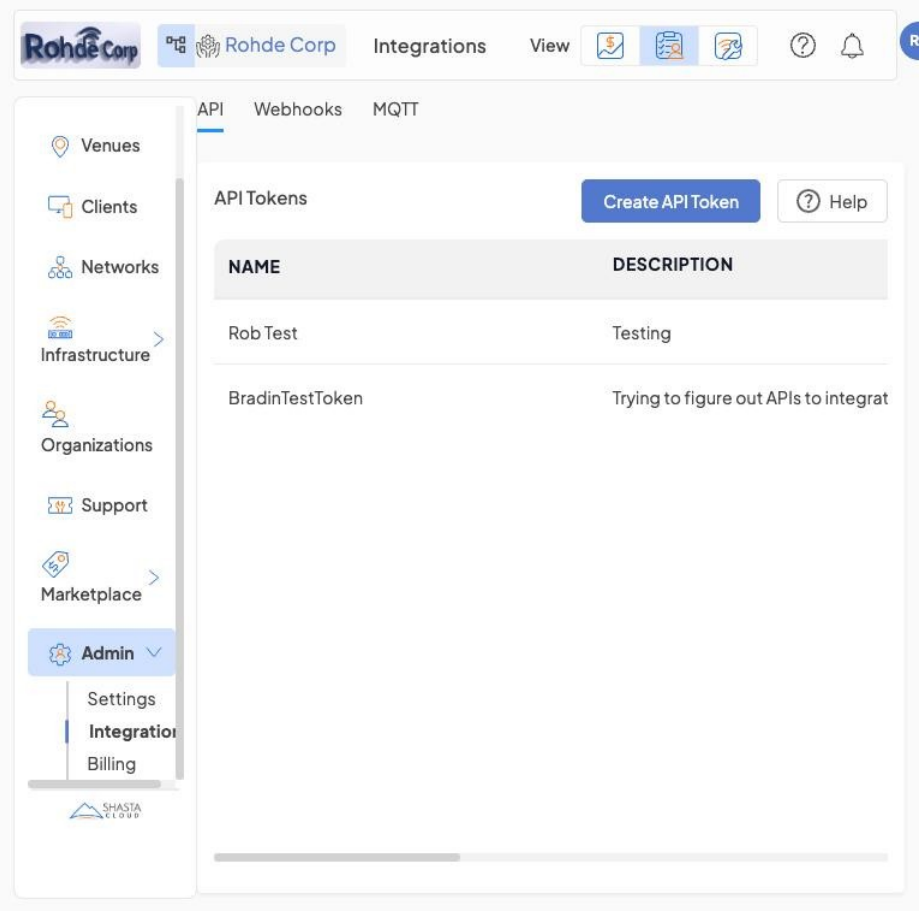


Figure 1 — Admin > Integrations > API tab showing the list of existing API tokens.

1.2 Click "Create API Token"

Click the blue Create API Token button in the upper right to open the token creation dialog.

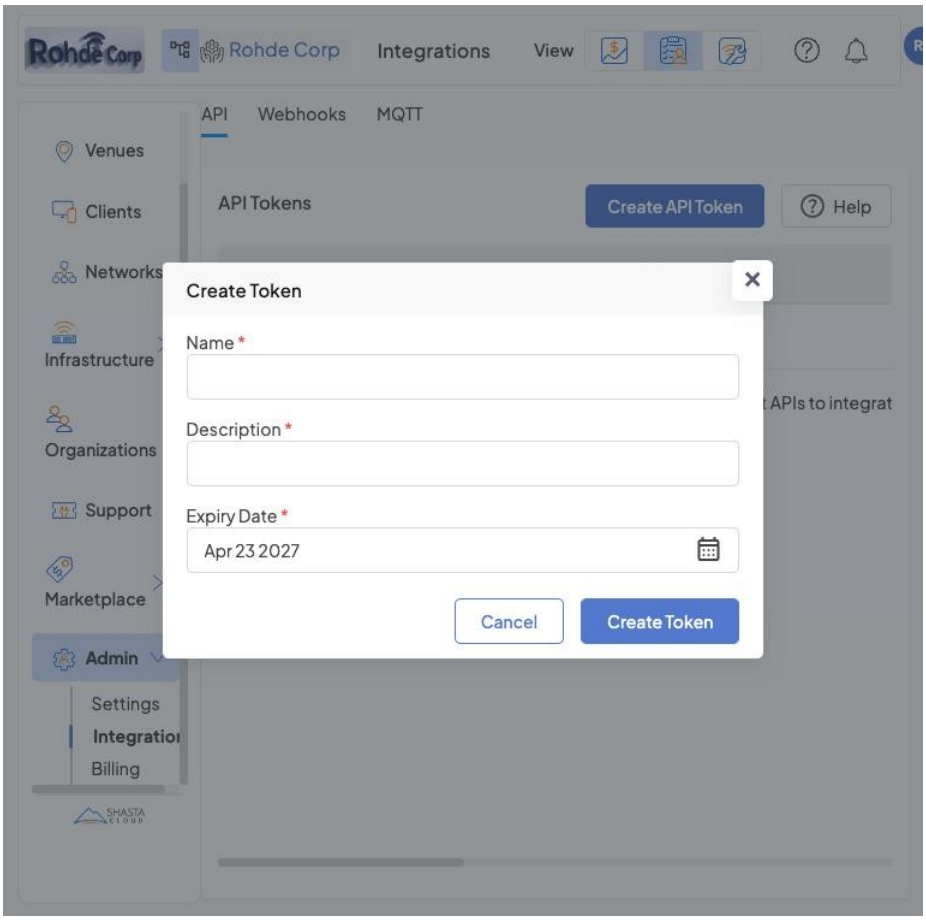


Figure 2 — The Create Token dialog with Name, Description, and Expiry Date fields.

1.3 Fill in Token Details

Complete the three required fields:

Field	Required	Description
Name	Required	A short, descriptive label for the token (e.g., "Production Integration", "Data Pipeline Token").
Description	Required	A brief explanation of what the token is used for (e.g., "API token for telemetry data access").
Expiry Date	Required	The date when the token will expire and be automatically disabled. Default is 1 year from today.

Click Create Token to generate the token.

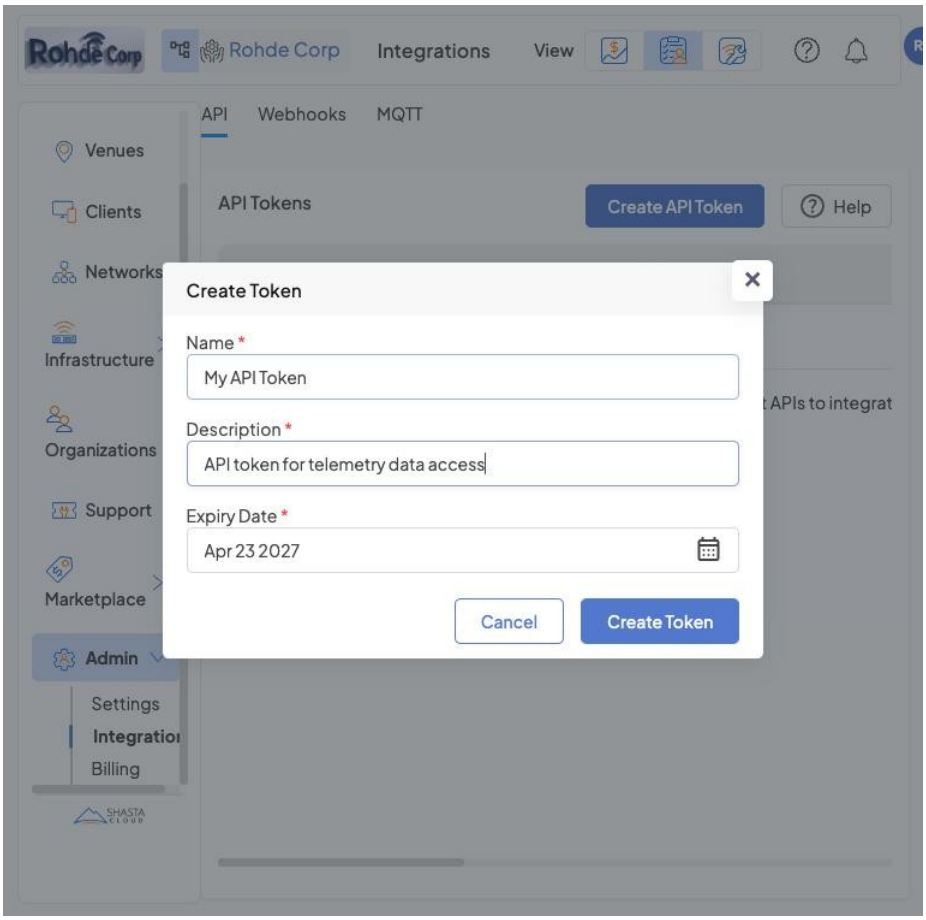


Figure 3 — Create Token dialog filled in with name, description, and expiry date.

1.4 Token Created Successfully

A green success notification confirms the token was created. The new token appears immediately in the table with status Enabled.

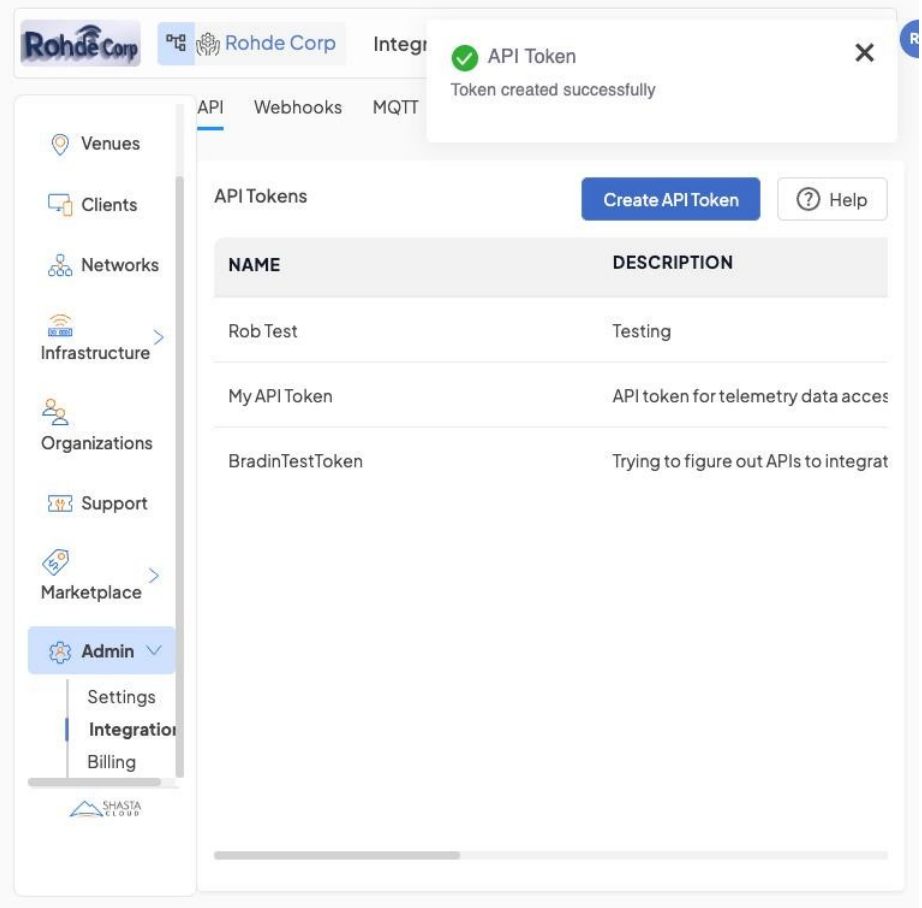


Figure 4 — "Token created successfully" confirmation notification.

1.5 Viewing the Full Token Table

The token table is wide and includes columns visible by scrolling right: Name, Description, Created Date, Expiry Date, Status, a Copy Token button, and an action menu (three dots).

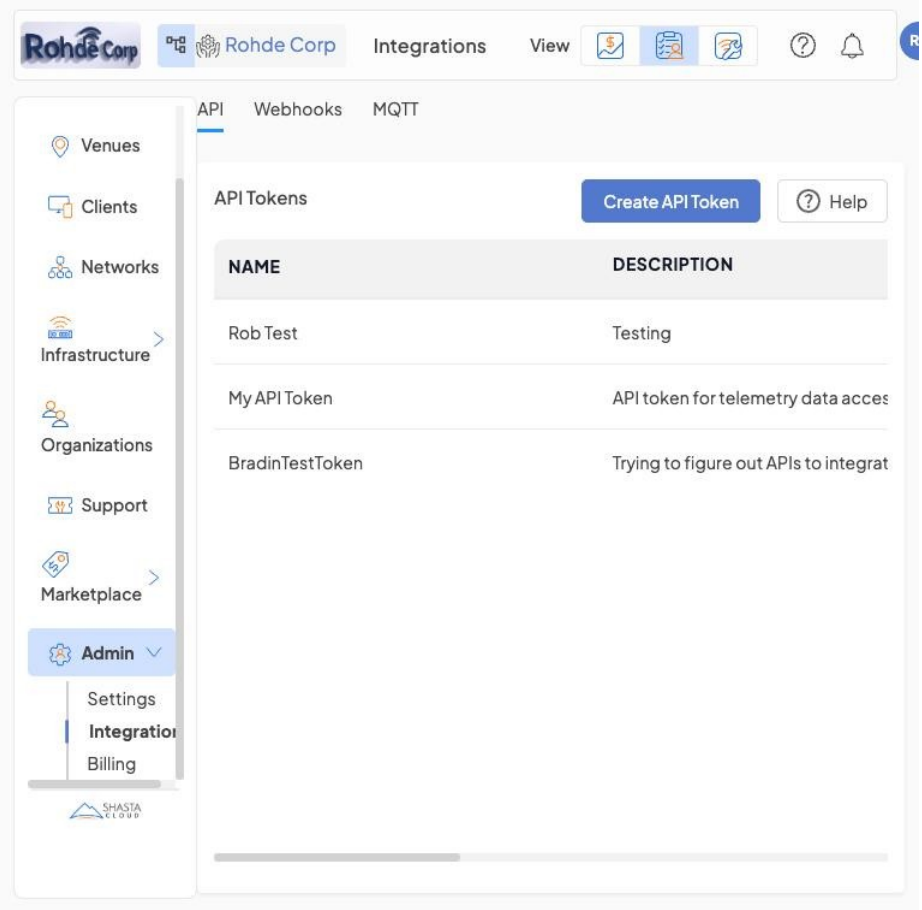


Figure 5 — Updated token list showing the newly created token entry.

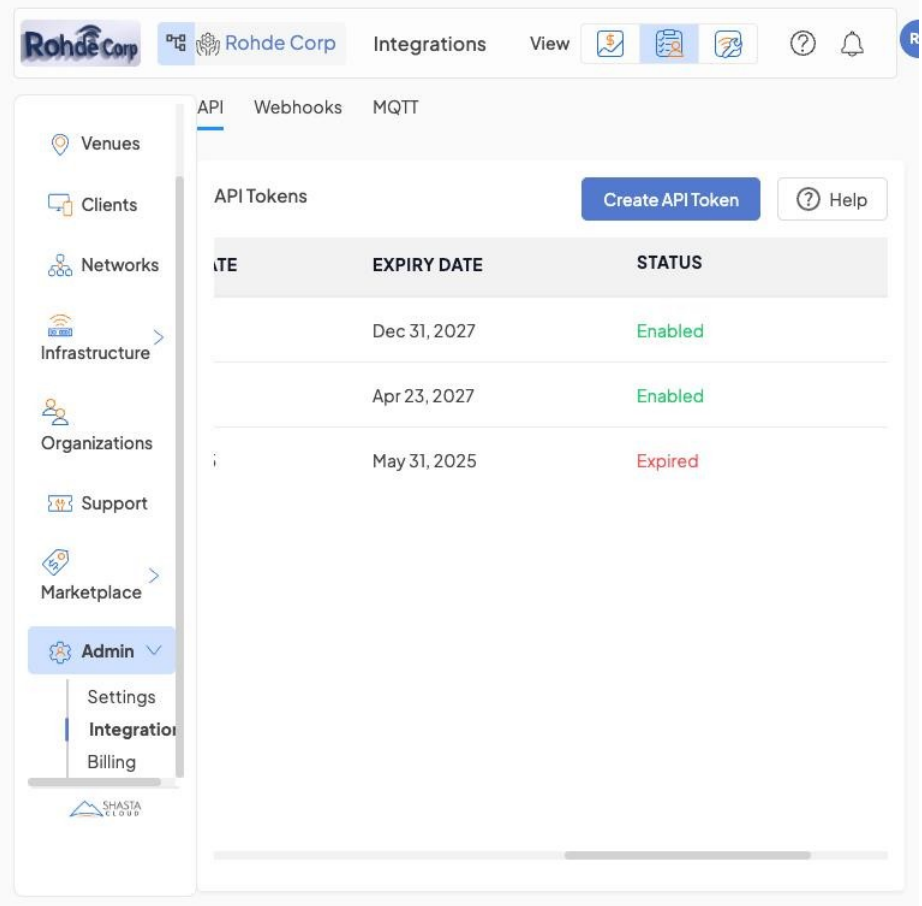


Figure 6 — Token table scrolled right to reveal Expiry Date, Status, and action buttons.

1.6 Copying Your Token

Click the Copy Token button next to any Enabled token to copy its value to your clipboard. This is the value you will use as the Bearer token in API calls.

> **Security:** Treat your API token like a password. Do not share it publicly or commit it to version control. If a token is compromised, use the Edit option to rotate it or Delete it immediately.

1.7 Token Actions Menu

Click the three-dot menu next to any token to access management options:

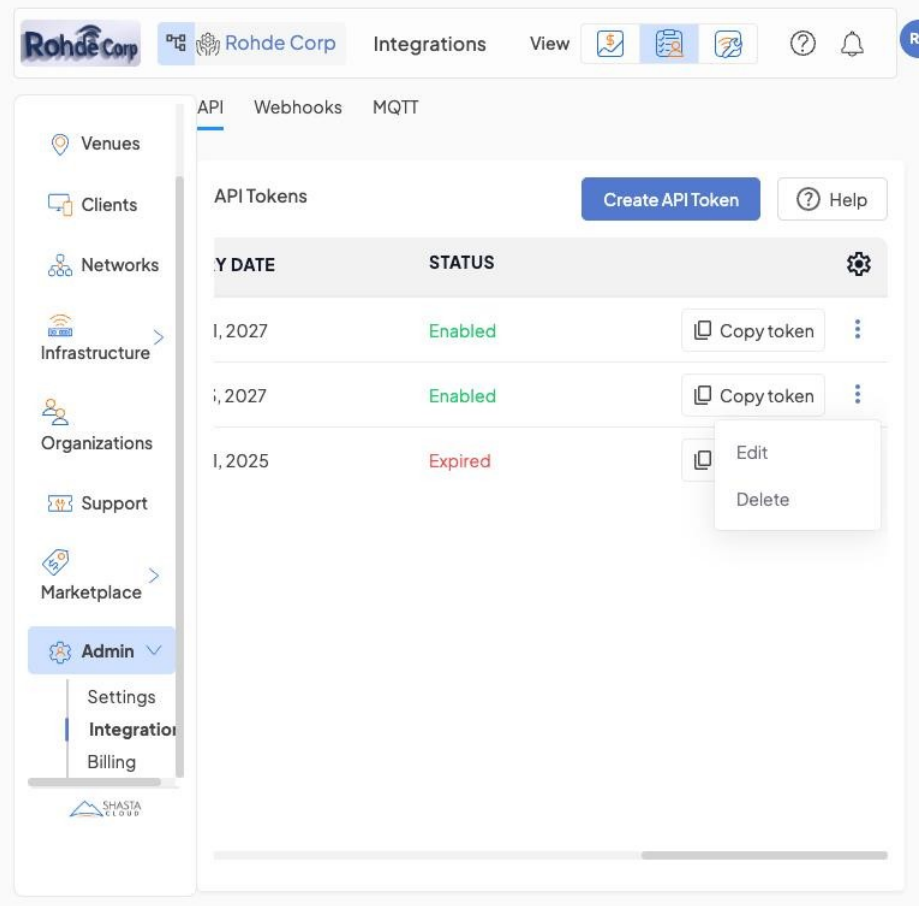


Figure 7 — Token action menu with Edit and Delete options.

Edit — Change the token's name, description, or expiry date.

Delete — Permanently remove the token. All API calls using this token will immediately fail.

Step 2 — Navigating to Integration Docs

The Integration Docs provide interactive API documentation powered by Swagger UI. From there, you can browse every available endpoint, read parameter descriptions, and execute live test calls directly in the browser.

2.1 Accessing Integration Docs

From the Admin > Integrations page, click the Help button in the upper right of the API tab. This navigates to the Integration Docs. You can also reach the docs from the platform's top-nav breadcrumb once you are inside the Integrations section.

> **Note:** The Integration Docs page is organized with three tabs: RESTful API, Webhooks, and MQTT. This guide focuses on the RESTful API tab.

2.2 Available API Services

The RESTful API tab displays a card-based grid of all available microservices. Each card shows the service name, version, authentication type (Private), and the OpenAPI Specification version.

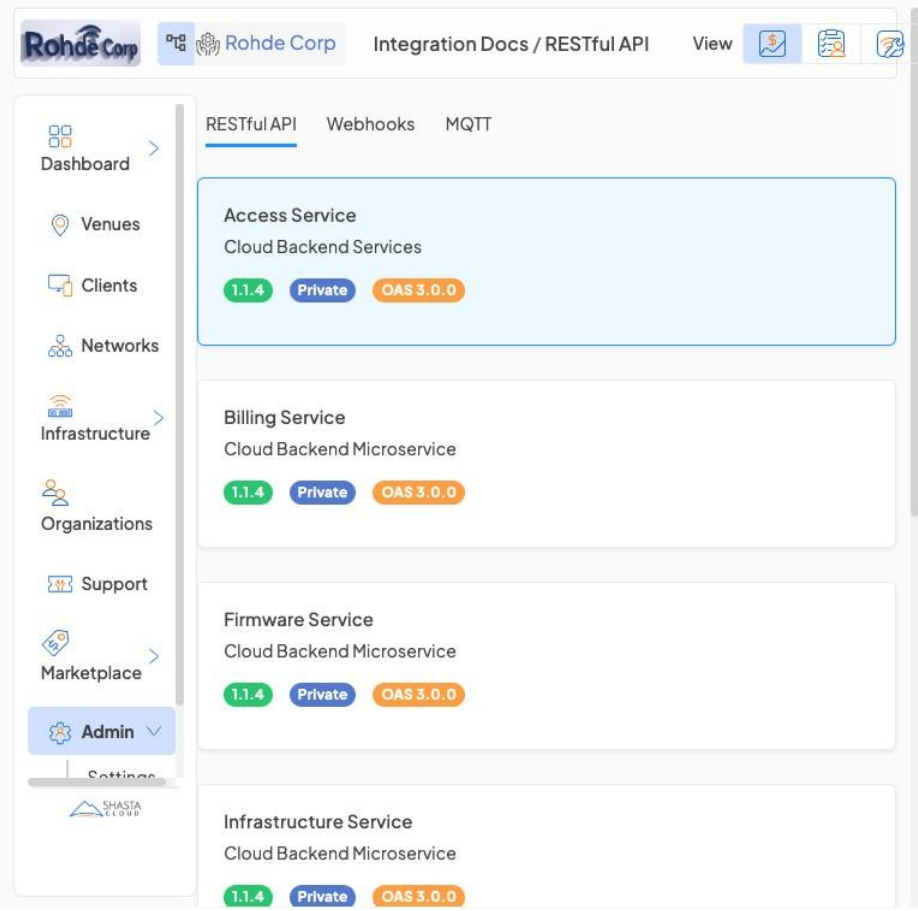


Figure 8 — Integration Docs main page showing Access, Billing, Firmware, and Infrastructure services.

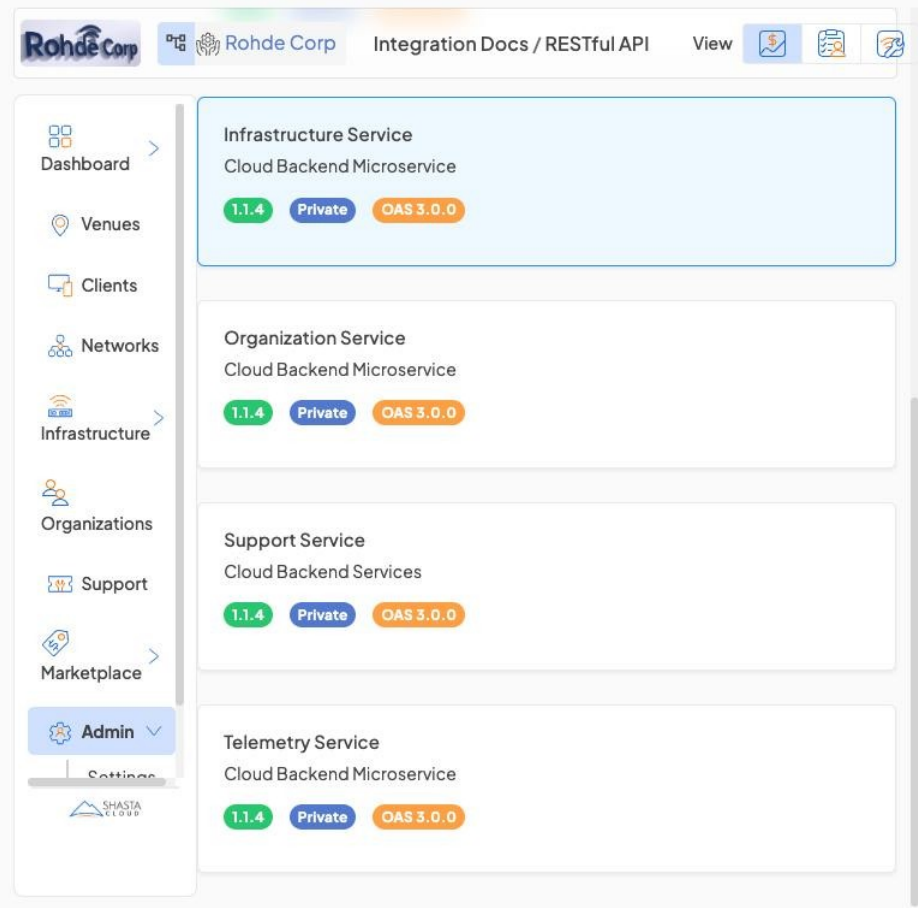


Figure 9 — Scrolling reveals Organization, Support, and Telemetry services.

2.3 API Service Catalog

The platform exposes seven distinct services, each responsible for a specific domain of functionality:

Service	Type	Primary Use Cases
Telemetry Service	Cloud Backend Microservice	Connected clients, events, infra stats, traffic, uptime, alerts
Access Service	Cloud Backend Services	Authentication, access control, identity management
Infrastructure Service	Cloud Backend Microservice	AP management, firmware, VLAN, provisioning
Organization Service	Cloud Backend Microservice	Org/venue hierarchy, user management
Billing Service	Cloud Backend	Subscription management, usage billing

Service	Type	Primary Use Cases
	Microservice	
Firmware Service	Cloud Backend Microservice	Firmware releases, update scheduling
Support Service	Cloud Backend Services	Tickets, knowledge base, feature requests

Click any service card to open its full Swagger UI documentation. For telemetry and client data, click Telemetry Service.

Step 3 — The Telemetry Service

The Telemetry Service is the primary API for accessing real-time and historical network data. It is a Cloud Backend Microservice running version 1.1.4.

Staging base URL: `https://api-stg.shastacloud.com/telemetry`

3.1 Telemetry Service Overview

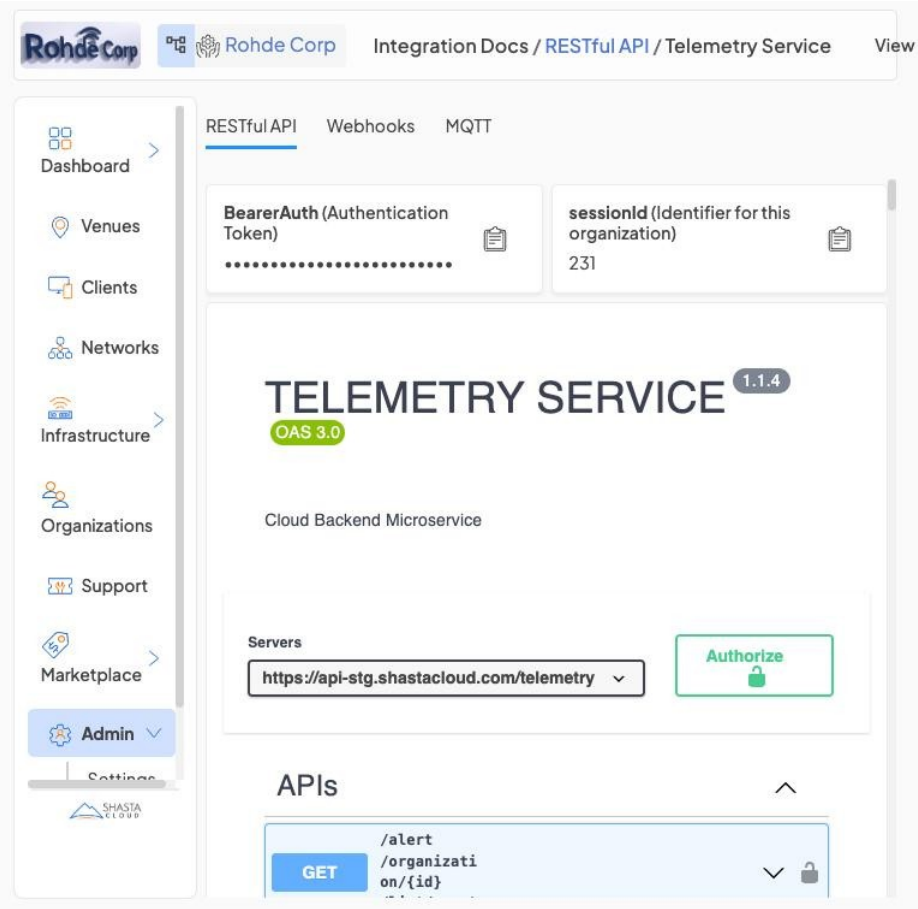


Figure 10 — Telemetry Service page showing BearerAuth field, Session ID, Server URL, and Authorize button.

The top of the Telemetry Service page exposes four controls:

BearerAuth (Authentication Token) — where your API token is stored for authenticated requests.

sessionId (Identifier for this organization) — pre-filled with your organization ID (e.g., 231).

Servers — the base URL dropdown (staging or production).

Authorize button — click to enter or change your authentication credentials.

3.2 Authorizing with Your API Token

Before executing API calls you must authorize the Swagger UI with your token. Click the green Authorize button to open the authorization dialog.

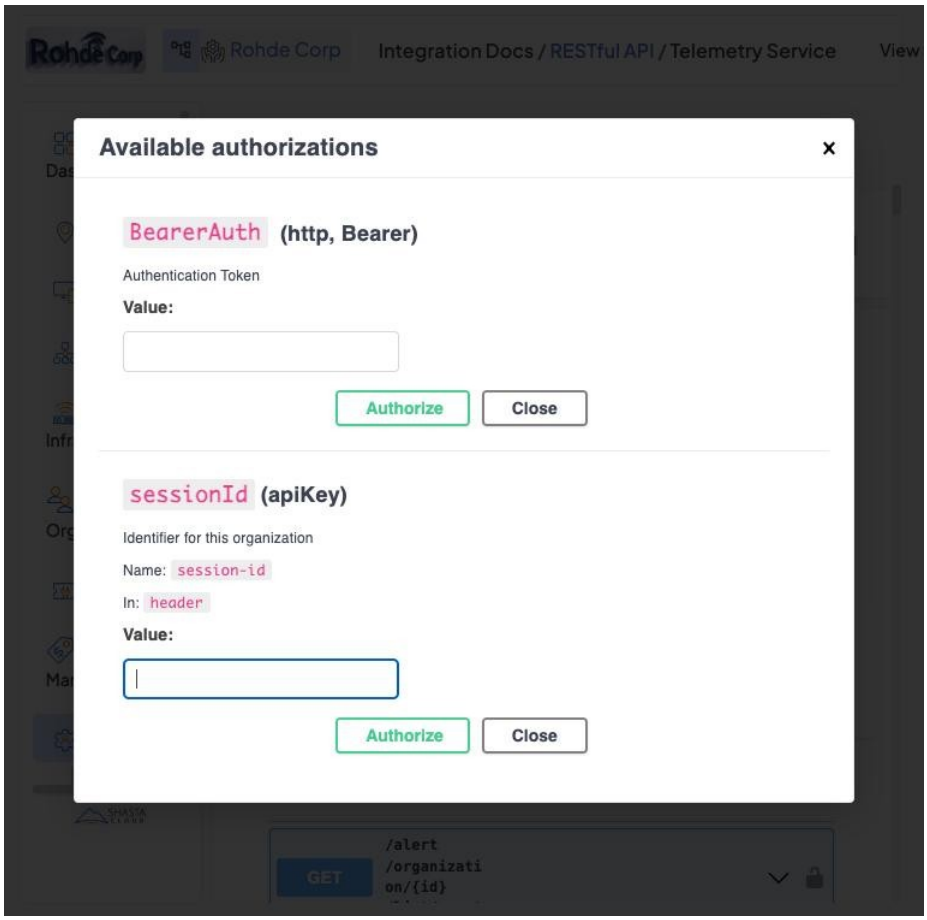


Figure 11 — Available Authorizations dialog with BearerAuth and sessionId fields.

In the authorization dialog:

1. In the BearerAuth **Value** field, paste the token you copied from Admin > Integrations.

2. Click Authorize next to BearerAuth to save the token for all requests.

3. Confirm the sessionId field contains your organization ID (e.g., 231). If not, enter it and click Authorize.

4. Click Close to dismiss the dialog.

> **Important:** Once authorized, the lock icons next to each endpoint appear unlocked and every "Try it out" request will automatically include your token in the Authorization header.

3.3 Telemetry Endpoint Categories

The Telemetry Service provides over 80 endpoints organized into logical groups:

Category	Endpoint Prefix	Description
Alerts	`/alert/*`	Organization-level network alarms and alerts
Events	`/events/*`	Device and venue event logs with date filtering
Infrastructure	`/infra/*`, `/infrastructure/*`	AP statistics, channel utilization, memory, traffic, uptime
Connected Clients	`/userdevice/*`	Real-time and historical client connection data
Organization	`/organization/*`	Dashboard metrics, venue stats, infra inventory
Traffic	`/traffic/*`	Per-device, per-venue, per-org traffic data
Uptime	`/uptime/*`	Device and venue uptime statistics
Timeline	`/timeline/*`	Historical device timeline (connects/disconnects)
Temperature	`/temperature/*`	Device temperature readings
Webhooks	`/webhook/*`	Webhook configuration and trends
MQTT	`/mqtt/*`	MQTT broker configuration

Step 4 — Example: Connected Clients API

The Connected Clients endpoints return devices currently (or recently) connected to your network. They live under the `/userdevice` endpoint group.

4.1 Available Client Endpoints

Method	Path	Purpose
GET	`/userdevice/organization/{id}/clients`	List all clients for an organization
GET	`/userdevice/organization/{id}/clients/count`	Count of clients for an organization
GET	`/userdevice/organization/{id}/clients/export`	Export client list as a downloadable file
GET	`/userdevice/venue/{id}/clients`	List clients for a specific venue
GET	`/userdevice/infrastructure/{id}/clients`	List clients connected to a specific AP
GET	`/userdevice/{mac}/details`	Detailed info for a specific client by MAC

4.2 Using GET /userdevice/organization/{id}/clients

This is the most common endpoint for retrieving connected clients across an entire organization. Click the endpoint row in Swagger UI to expand it.

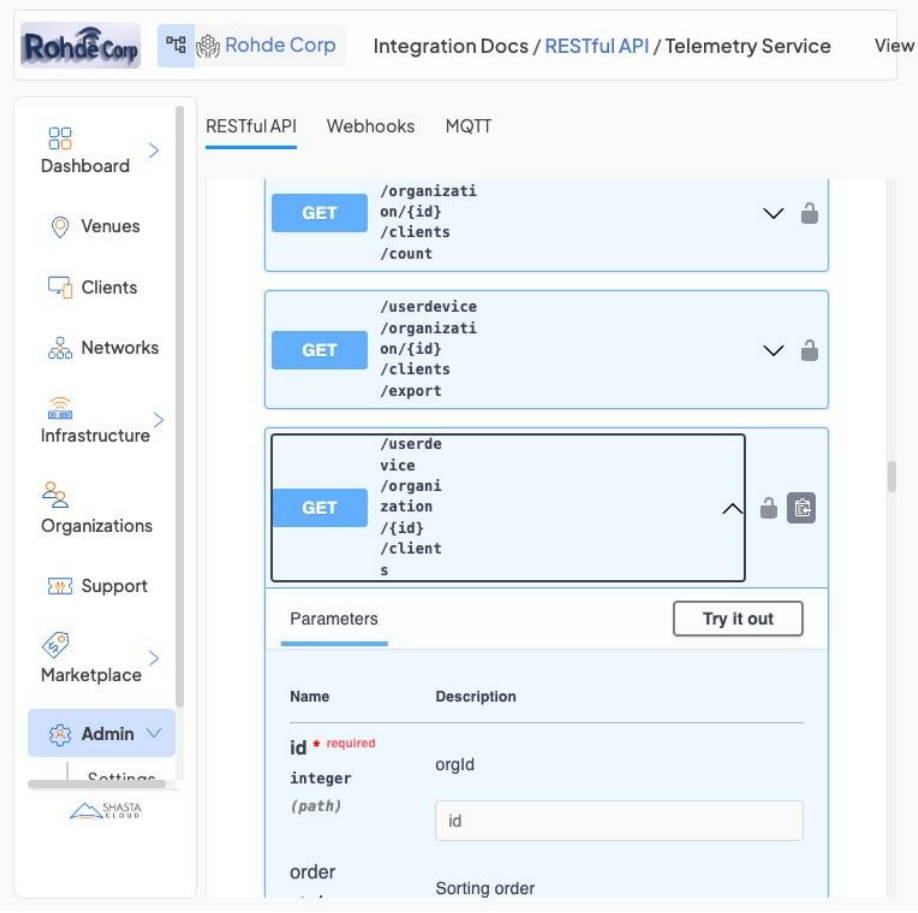


Figure 12 — GET /userdevice/organization/{id}/clients expanded, showing the Parameters tab and Try it out button.

4.3 Parameters

The clients endpoint supports rich filtering. Click Try it out to enable interactive parameter input.

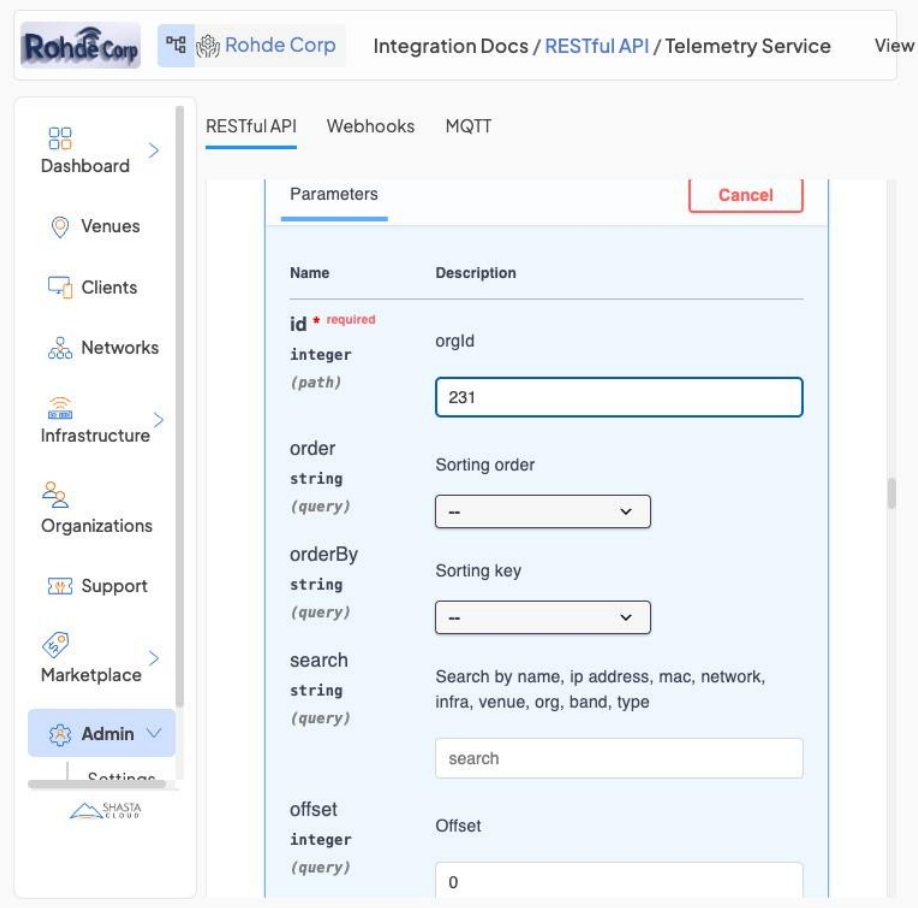


Figure 13 — "Try it out" mode with parameters (id=231, clientStatus=connected, clientType=wireless).

Parameter	Type	Required	Description
`id`	integer (path)	**Yes**	Organization ID (orgId) — e.g., 231
`order`	string (query)	No	Sorting order: asc or desc
`orderBy`	string (query)	No	Sorting key (e.g., name, rssi, connected)
`search`	string (query)	No	Search by name, IP, MAC, network, infra, venue, org, band, type
`offset`	integer (query)	No	Pagination offset (default: 0)
`limit`	integer (query)	No	Max results per page (default: 10)
`children`	boolean (query)	No	Include clients from child organizations (default: false)
`clientStatus`	string (query)	No	Filter by status: connected, disconnected, or all

Parameter	Type	Required	Description
`clientType`	string (query)	No	Filter by type: wireless or wired
`venueIds`	string (query)	No	Filter by specific venue IDs (comma-separated)
`infraIds`	string (query)	No	Filter by specific AP IDs (comma-separated)
`ssid`	string (query)	No	Filter by SSID name(s), comma-separated
`band`	string (query)	No	Filter by radio band (comma-separated)

4.4 Executing the Request

Fill in the id field with your Organization ID, set any desired filters, and click the blue Execute button.

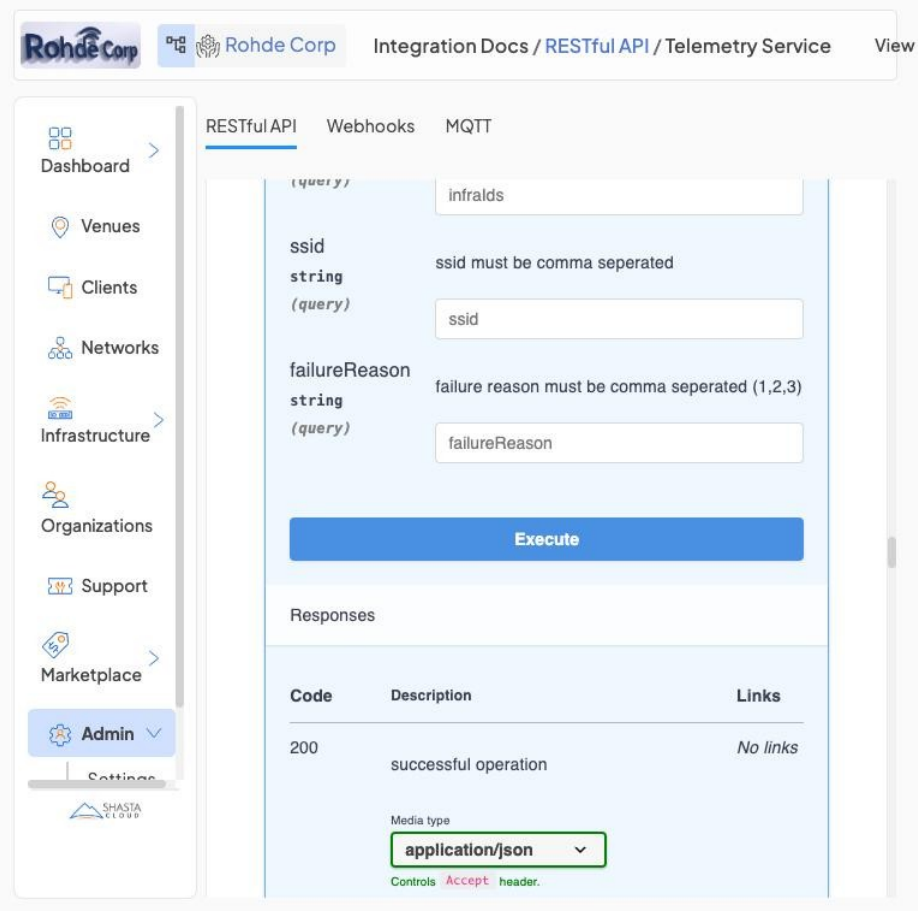


Figure 14 — Execute button and Responses section showing 200 Successful Operation.

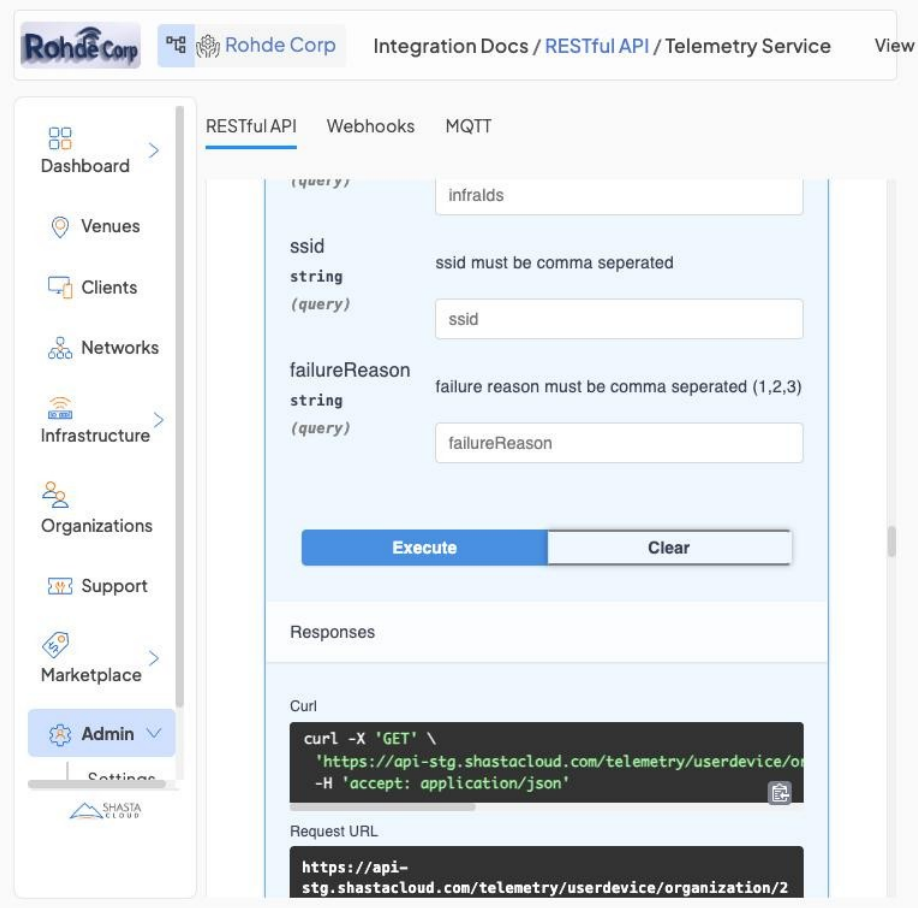


Figure 15 — After clicking Execute, the equivalent curl command and full Request URL are displayed.

Swagger UI prints the equivalent curl command and the full request URL for reference:

```
curl -X 'GET' \
  'https://api-stg.shastacloud.com/telemetry/userdevice/organization/231/
clients\
   ?
offset=0&limit=10&children=false&clientStatus=connected&clientType=wireless' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer YOUR_API_TOKEN_HERE'
```

4.5 Authentication Required (401 Error)

If you execute a request without first authorizing — or with an expired token — the API returns a 401 Unauthorized response.

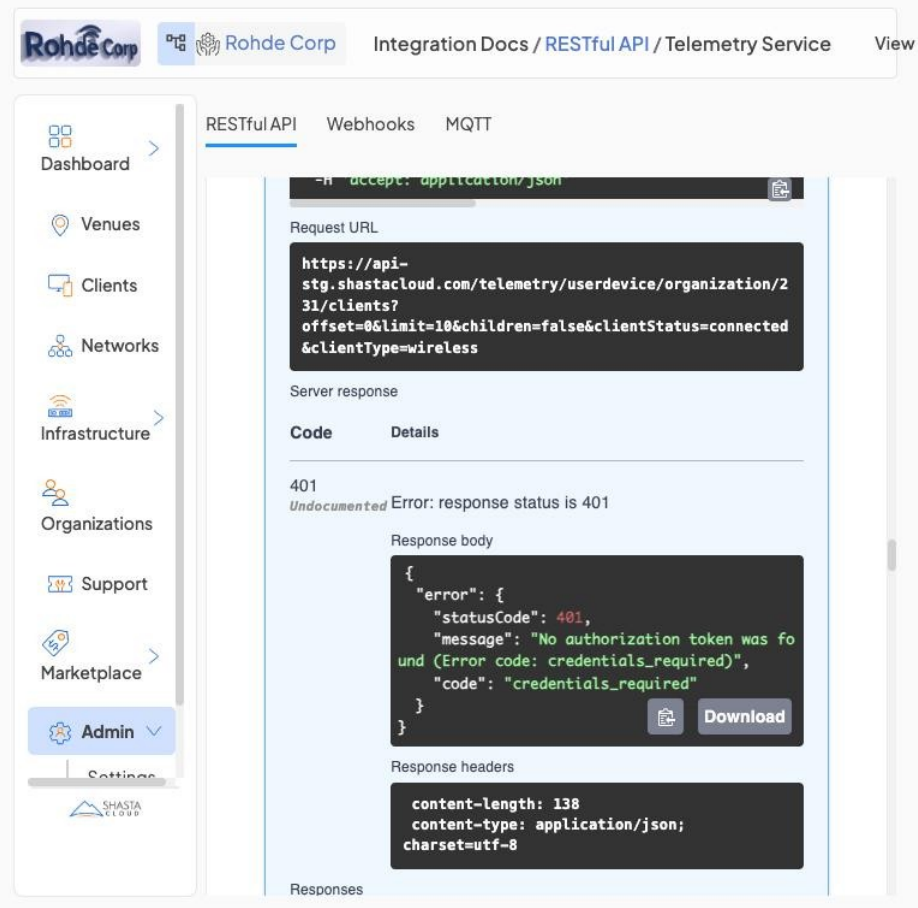


Figure 16 — 401 response: "No authorization token was found (credentials_required)".

```
{
  "error": {
    "statusCode": 401,
    "message": "No authorization token was found (Error code:
credentials_required)",
    "code": "credentials_required"
  }
}
```

> **Warning:** Always click Authorize and enter your Bearer token before executing API calls. Without a valid token, every request will return 401 Unauthorized.

Step 5 — Example: Events API

The Events endpoints return a log of network events for devices or venues — client associations, disassociations, authentication failures, roaming events, and AP state changes.

5.1 Available Events Endpoints

Method	Path	Purpose
GET	`/events/venue/{id}`	All events for a venue in a date range
GET	`/events/venue/{id}/count`	Count of events for a venue
GET	`/events/venue/{id}/export`	Export venue events as a file
GET	`/events/device/{macAddress}`	Events for a specific AP by MAC address
GET	`/userdevice/{mac}/events`	Events for a specific client device
GET	`/events-description`	Descriptions of all event types

5.2 Using GET /events/venue/{id}

This endpoint retrieves all events for a specific venue within a supplied date range. Click the endpoint row in Swagger UI to expand it.

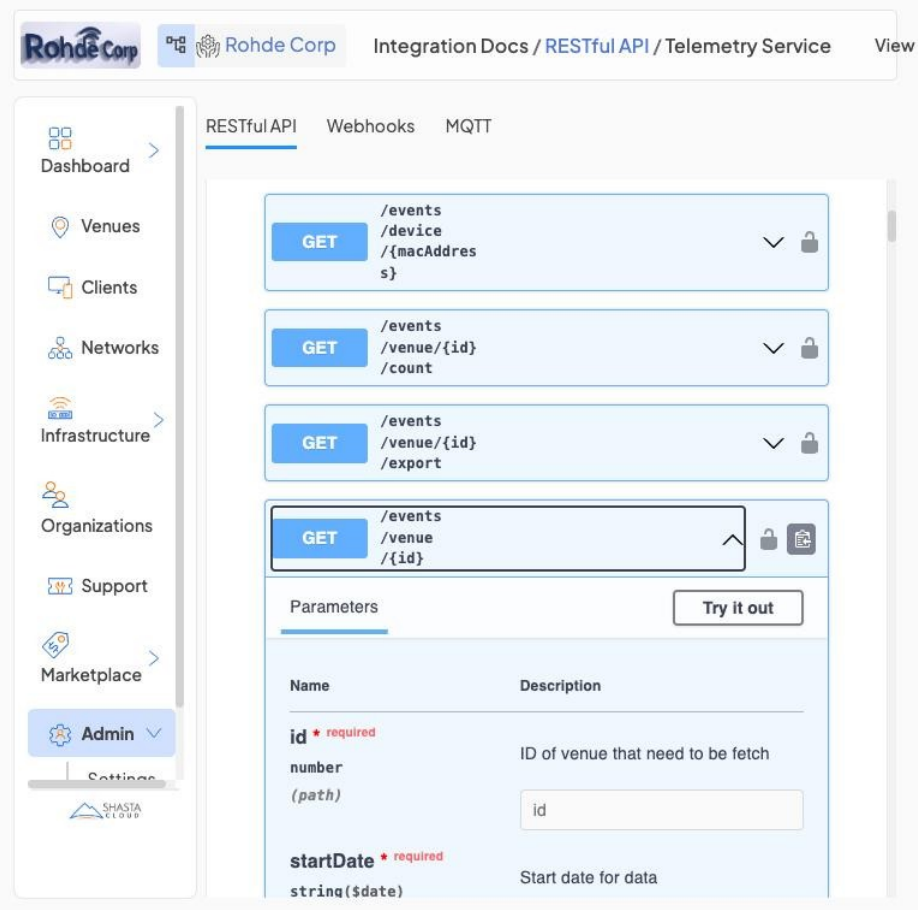


Figure 17 — GET /events/venue/{id} expanded, showing the Parameters tab and Try it out button.

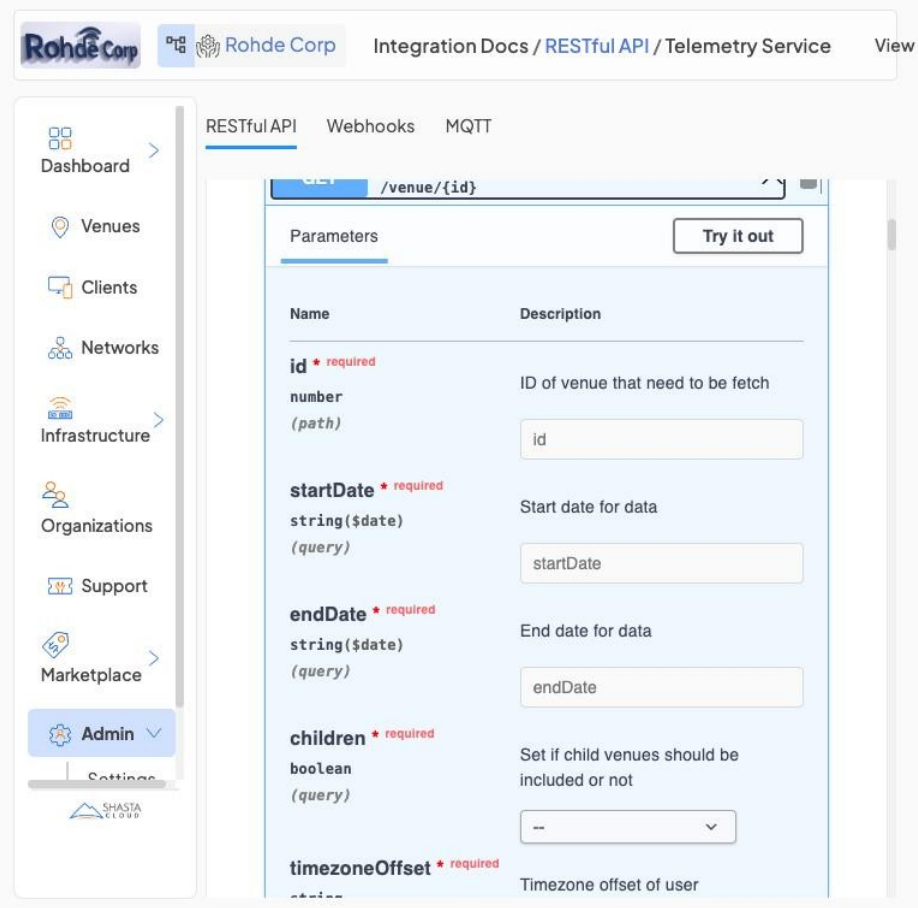


Figure 18 — Events endpoint parameters: venue ID, start/end date range, children flag, and timezone offset.

5.3 Parameters

Parameter	Type	Required	Description
`id`	number (path)	**Yes**	Numeric venue ID
`startDate`	string($date) (query)	**Yes**	Start date for data retrieval (YYYY-MM-DD)
`endDate`	string($date) (query)	**Yes**	End date for data retrieval (YYYY-MM-DD)
`children`	boolean (query)	**Yes**	Include events from child venues
`timezoneOffset`	string (query)	**Yes**	Timezone offset (e.g., -08:00)

5.4 Example Request

To retrieve all events for venue 339 for the past week:

```
curl -X 'GET' \
  'https://api-stg.shastacloud.com/telemetry/events/venue/339\
   ?startDate=2026-04-16&endDate=2026-04-23&children=false&timezoneOffset=-
07:00' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer YOUR_API_TOKEN_HERE'
```

5.5 Event Types

To understand what event codes mean, call `GET /events-description`. It returns a glossary of every event type with human-readable names. Common categories include:

Event Category	Examples
Client Association	Client connected, client disconnected, roaming
Authentication	Auth success, auth failure, 802.1X events
AP State	AP online, AP offline, AP rebooted
Network	VLAN changes, channel changes, interference events
Security	Rogue AP detected, IDS/IPS events

Step 6 — Using the API with curl and Code

Once you have a token, you can make authenticated calls directly from the command line or integrate them into your applications.

6.1 Authentication Header

Every Telemetry API request must include the Bearer token and the session ID (organization ID):

```
# Required headers for all Telemetry API calls
-H 'Authorization: Bearer YOUR_API_TOKEN_HERE'
-H 'session-id: 231'
-H 'accept: application/json'
```

6.2 Example — List Connected Wireless Clients

```
# Get all currently connected wireless clients for org 231
curl -X GET \
  'https://api-stg.shastacloud.com/telemetry/userdevice/organization/231/
clients' \
  --url-query 'clientStatus=connected' \
  --url-query 'clientType=wireless' \
  --url-query 'limit=50' \
  --url-query 'offset=0' \
  -H 'Authorization: Bearer YOUR_API_TOKEN' \
  -H 'session-id: 231' \
  -H 'accept: application/json'
```

6.3 Example — Get Client Count for Organization

```
# Get the count of connected clients across the organization
curl -X GET \
  'https://api-stg.shastacloud.com/telemetry/userdevice/organization/231/
clients/count' \
  -H 'Authorization: Bearer YOUR_API_TOKEN' \
  -H 'session-id: 231' \
  -H 'accept: application/json'
```

6.4 Example — Get Events for a Venue (Last 7 Days)

```
# Get all events for venue 339 for the past week
```

```
curl -X GET \
  'https://api-stg.shastacloud.com/telemetry/events/venue/339' \
  --url-query 'startDate=2026-04-16' \
  --url-query 'endDate=2026-04-23' \
  --url-query 'children=false' \
  --url-query 'timezoneOffset=-07:00' \
  -H 'Authorization: Bearer YOUR_API_TOKEN' \
  -H 'session-id: 231' \
  -H 'accept: application/json'
```

6.5 Example — Python Integration

```
import requests

API_BASE = "https://api-stg.shastacloud.com/telemetry"
TOKEN    = "YOUR_API_TOKEN_HERE"
ORG_ID   = 231

headers = {
    "Authorization": f"Bearer {TOKEN}",
    "session-id":    str(ORG_ID),
    "accept":        "application/json",
}

# Get connected clients
response = requests.get(
    f"{API_BASE}/userdevice/organization/{ORG_ID}/clients",
    headers=headers,
    params={
        "clientStatus": "connected",
        "clientType":   "wireless",
        "limit":        100,
        "offset":       0,
    },
)

data = response.json()
print(f"Connected clients: {len(data.get('data', []))}")
for client in data.get("data", []):
    print(f"  {client['name']} - {client['ip']} - RSSI: {client.get('rssi',
'N/A')} dBm")
```

> **Tip:** Use Swagger UI's "Try it out" feature to construct your call first. After clicking Execute, copy the displayed curl command — it will have all parameters properly formatted and can be used directly in your terminal or adapted for any language.

Step 7 — API Token Management Best Practices

Properly managing API tokens is important for maintaining the security and reliability of your integrations.

7.1 Token Status

Status	Description	Action
Enabled	Token is active and can authenticate API calls	Use for API access; monitor expiry date
Expired	Token has passed its expiry date and will return 401 errors	Delete old token, create new one, update integrations

7.2 Security Best Practices

Use descriptive names — make token names meaningful so you can identify which integration uses which token (e.g., "Dashboard Integration — Production", "Reporting Pipeline").

Set appropriate expiry dates — shorter-lived tokens are more secure but require more frequent rotation. Align expiry with your maintenance schedule.

Store tokens securely — use environment variables, secrets managers (AWS Secrets Manager, HashiCorp Vault), or encrypted configuration files rather than hardcoding tokens.

Monitor expiry — set calendar reminders before tokens expire to avoid service interruptions.

One token per integration — separate tokens per system or script make it easy to revoke access to a single integration without affecting others.

Delete unused tokens — regularly review and delete tokens that are no longer in use to reduce the attack surface.

7.3 Rotating a Token

1. Create a new token with a new name and sufficient expiry date.

2. Copy the new token value.

3. Update all systems, scripts, and integrations using the old token to use the new one.

4. Test that integrations are working with the new token.

5. Delete the old token once all systems have been updated.

> **Important:** Deleting a token immediately invalidates it. All API calls using that token will

> return 401 errors. Always update integrations before deleting the old token.

Quick Reference Summary

A condensed reference of the key configuration values and endpoints covered in this guide.

Base URL

`https://api-stg.shastacloud.com/telemetry`

Staging environment. Production URL is provided separately.

Authentication

```
Authorization: Bearer <token>
session-id: <orgId>
```

Both headers required on every request.

Token Location

Admin > Integrations > API

Create, edit, copy, or delete tokens. Copy Token puts the value on your clipboard.

Common Endpoints

```
GET /userdevice/organization/{id}/clients
GET /events/venue/{id}
GET /events-description
```

Tip: For the complete list of endpoints and their schemas, open the Telemetry Service in Integration Docs and browse the Swagger UI interactively.