

Hotspot Locations

Endpoints Summary

Method	Path	Swagger
GET	/hotspot_locations/	Swagger ↗

“ The Hotspot Locations API provides access to physical location data for hotspot devices within your organization. This endpoint allows you to retrieve detailed information about locations including geographical coordinates, addresses, and organizational details, making it essential for location-based analytics, mapping applications, and device management systems.

Base URL: `https://gate.zequenze.com/api/v1`

Authentication: All endpoints require a Bearer token:

```
Authorization: Bearer <your-api-token>
```

Overview

The Hotspot Locations API enables you to manage and retrieve location data for hotspot devices across your organization's network infrastructure. This API is particularly useful for:

- **Network Management:** Track and organize physical locations where hotspot devices are deployed
- **Geographic Analytics:** Analyze network usage patterns across different locations and regions
- **Mapping Applications:** Display hotspot locations on interactive maps for administrative dashboards
- **Location-Based Services:** Provide location-aware features and services to end users
- **Compliance & Reporting:** Generate location-based reports for regulatory compliance or business intelligence

The API supports filtering by coordinates, change tracking for synchronization purposes, and comprehensive location metadata including addresses, postal codes, and geographic coordinates. All location data is scoped to your organization and includes both active and inactive locations for complete lifecycle management.

Endpoints

GET /hostspot_locations/

Description: Retrieves a paginated list of all hotspot locations within your organization. This endpoint supports various filtering options to help you find specific locations based on coordinates availability, recent changes, or other criteria. Use this endpoint to populate location dropdowns, generate maps, or synchronize location data with external systems.

Use Cases:

- Building a map interface showing all hotspot locations with coordinates
- Synchronizing location data with external systems using the last_change filter
- Generating reports of active vs inactive locations
- Populating location selection interfaces for device management

Full URL Example:

```
https://gate.zequenze.com/api/v1/hostspot_locations/?have_coordinates=true&limit=50&offset=0
```

Parameters:

Parameter	Type	In	Required	Description
last_change_gte	string	query	No	Filter locations modified since a specific date/time. Use ISO format: <code>2000-01-01</code> , <code>2000-01-01 00:01:00</code> , or <code>2000-01-01 00:01:00+00:00</code>
limit	integer	query	No	Number of results to return per page (default pagination applies)
offset	integer	query	No	The starting index for results (used for pagination)

Parameter	Type	In	Required	Description
have_coordinates	boolean	query	No	When <code>true</code> , returns only locations with latitude and longitude values set

cURL Example:

```
curl -X GET
"https://gate.zequenze.com/api/v1/hostspot_locations/?have_coordinates=true&limit=10" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json"
```

Example Response:

```
{
  "count": 156,
  "next": "https://gate.zequenze.com/api/v1/hostspot_locations/?limit=10&offset=10",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "Downtown Coffee Shop",
      "short_name": "downtown-coffee",
      "is_active": true,
      "description": "High-traffic location in downtown business district",
      "organization_id": "org_12345",
      "thumbnail": "https://gate.zequenze.com/media/thumbnails/location_1.jpg",
      "address": "123 Main Street",
      "reference": "Building A, Ground Floor",
      "city": "San Francisco",
      "postal_code": "94105",
      "state": "California",
      "region": "Bay Area",
      "country_code": "US",
      "latitude": "37.7749",
      "longitude": "-122.4194",
      "created": "2024-01-15T10:30:00Z",
      "last_change": "2024-03-10T14:22:00Z"
    },
    {
```

```
"id": 2,
"name": "Airport Terminal B",
"short_name": "airport-terminal-b",
"is_active": true,
"description": "International departures terminal hotspot",
"organization_id": "org_12345",
"thumbnail": null,
"address": "San Francisco International Airport",
"reference": "Terminal B, Gate Area",
"city": "San Francisco",
"postal_code": "94128",
"state": "California",
"region": "Bay Area",
"country_code": "US",
"latitude": "37.6213",
"longitude": "-122.3790",
"created": "2024-02-01T08:15:00Z",
"last_change": "2024-03-05T16:45:00Z"
}
]
}
```

Response Codes:

Status	Description
200	Success - Returns the paginated list of hotspot locations
401	Unauthorized - Invalid or missing Bearer token
403	Forbidden - Token valid but lacks permission to access locations
422	Unprocessable Entity - Invalid parameter format (e.g., malformed date)

Common Use Cases

Use Case 1: Building a Location Map Dashboard

Retrieve all locations with coordinates to display on an interactive map for network administrators.

```
https://gate.zequenze.com/api/v1/hostspot_locations/?have_coordinates=true&is_active=true
```

Use Case 2: Syncing Location Data

Synchronize location changes since your last update using the timestamp filter for data consistency.

```
https://gate.zequenze.com/api/v1/hostspot_locations/?last_change__gte=2024-03-01T00:00:00Z
```

Use Case 3: Location Selection Interface

Load locations with pagination for device assignment or configuration interfaces.

```
https://gate.zequenze.com/api/v1/hostspot_locations/?limit=25&offset=0
```

Use Case 4: Geographic Analytics

Filter locations by region or coordinates to analyze network coverage and usage patterns.

```
https://gate.zequenze.com/api/v1/hostspot_locations/?have_coordinates=true&limit=100
```

Use Case 5: Inventory Management

Retrieve all locations (active and inactive) for comprehensive device deployment planning.

```
https://gate.zequenze.com/api/v1/hostspot_locations/?limit=50
```

Best Practices

- **Use Pagination Effectively:** Always implement pagination for large datasets. Start with reasonable page sizes (25-50 items) and adjust based on your application's performance requirements.
- **Filter by Coordinates:** When building mapping applications, use `have_coordinates=true` to avoid processing locations without geographic data.
- **Implement Change Tracking:** Use the `last_change__gte` parameter for efficient synchronization. Store the timestamp of your last successful sync and use it in subsequent

requests.

- **Cache Location Data:** Location information typically changes infrequently, making it ideal for caching. Consider implementing a cache invalidation strategy based on the `last_change` field.
 - **Handle Missing Data Gracefully:** Not all locations may have complete address information or coordinates. Design your application to handle optional fields appropriately.
 - **Monitor Rate Limits:** Implement proper error handling and retry logic for API rate limits, especially when processing large numbers of locations.
 - **Validate Geographic Data:** When using latitude/longitude values, validate that they fall within expected ranges and handle edge cases for locations near poles or the international date line.
-
-

Revision #4

Created 2026-02-04 05:14:56 UTC by ipena@zequenze.com

Updated 2026-02-11 03:18:25 UTC by ipena@zequenze.com