Device App Connection Log

Endpoints Summary

Method	Path	Swagger
`POST`	`/device_app_connection_log/`	Swagger ↗

> The Device App Connection Log API enables applications to register and track network connection details from devices. This endpoint is essential for monitoring network usage patterns, connection quality metrics, and geographic location data across different network types including WiFi, Mobile, Bluetooth, and Ethernet connections.

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

Authentication: All endpoints require a Bearer token:

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

Overview

The Device App Connection Log API provides a centralized way to collect and store detailed network connection information from devices across your organization. This API is particularly valuable for:

- **Network Analytics**: Track connection patterns, network performance, and usage statistics across different locations and network types
- **Quality Monitoring**: Collect signal strength, link speed, and connection quality metrics to identify network issues
- **Geographic Tracking**: Monitor device connectivity across different locations using latitude/longitude data
- **Security Analysis**: Track network security types and authentication methods to ensure compliance with security policies
- **Roaming Detection**: Identify when devices are operating outside their home networks

The connection logs capture comprehensive network details including network identification (BSSID, Cell ID, MCC/MNC), performance metrics (RSSI, SINR, link speed), data usage (inbound/outbound bytes), and location information. This data is essential for network administrators, IT teams, and developers building location-aware or network-dependent applications.

Endpoints

POST /device_app_connection_log/

Description: Registers a comprehensive network connection log entry with detailed network characteristics, performance metrics, and location data. This endpoint accepts connection data from various network types and stores it for analytics and monitoring purposes.

Use Cases:

- Mobile applications reporting network connectivity status and performance
- IoT devices logging connection quality for network optimization
- Enterprise applications tracking employee device connectivity across locations
- Network monitoring tools collecting performance data from field devices

Full URL Example:

```
https://gate.zequenze.com/api/v1/device_app_connection_log/
```

Parameters:

Parameter	Type	In	Required	Description
data	string	body	Yes	JSON string containing the network connection log data

Request Body Schema:

Field	Type	Required	Description
network_name	string	No	Name of the current network (e.g., "Office WiFi", "Guest Network")
network_type	string	Yes	Network type: 'wi' (WiFi), 'mo' (Mobile), 'bl' (Bluetooth), 'et' (Ethernet)

Field	Type	Required	Description
network_security	string	Yes	Security type: 'opn' (Open), 'wep' (WEP), 'wpa' (WPA), 'wp2' (WPA2), '3gp' (3GPP)
network_mccmnc	string	No	Mobile network MCC/MNC code (e.g., "31026" for T-Mobile US)
frequency	number	No	Wireless network frequency in GHz (e.g., 2.4, 5.0)
authentication	string	No	Authentication method: 'non', 'cpo' (Captive portal), 'wis' (WISPr), 'psk', 'rad' (RADIUS), 'eap', '3gp'
bssid	string	No	Wireless network BSSID (MAC address of access point)
cellid	string	No	Mobile network Cell ID for cellular connections
roaming	boolean	No	Whether device is roaming outside its home network
latitude	string	No	Device latitude in decimal format
longitude	string	No	Device longitude in decimal format
inbytes	integer	No	Bytes received during this connection session
outbytes	integer	No	Bytes transmitted during this connection session
link_speed	number	No	Connection speed in Mbps as reported by network adapter
quality_metric	string	No	Quality measurement type: 'rssi' (RSSI) or 'sinr' (SINR)
quality_value	number	No	Signal quality value corresponding to the metric type

cURL Example:

```
curl -X POST "https://gate.zequenze.com/api/v1/device_app_connection_log/" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
```

```
    "network_name": "Office WiFi Main",
    "network_type": "wi",
    "network_security": "wp2",
    "frequency": 5.0,
    "authentication": "eap",
    "bssid": "00:1a:2b:3c:4d:5e",
    "roaming": false,
    "latitude": "40.7128",
    "longitude": "-74.0060",
    "inbytes": 1048576,
    "outbytes": 524288,
    "link_speed": 150.0,
    "quality_metric": "rssi",
    "quality_value": -45
  }'
```

Example Response:

```
{
  "network_name": "Office WiFi Main",
  "network_type": "wi",
  "network_security": "wp2",
  "network_mccmnc": null,
  "frequency": 5.0,
  "authentication": "eap",
  "bssid": "00:1a:2b:3c:4d:5e",
  "cellid": null,
  "roaming": false,
  "latitude": "40.7128",
  "longitude": "-74.0060",
  "inbytes": 1048576,
  "outbytes": 524288,
  "link_speed": 150.0,
  "quality_metric": "rssi",
  "quality_value": -45.0
}
```

Response Codes:

Status	Description

201	Created - Connection log successfully registered
400	Bad Request - Invalid data format or missing required fields
401	Unauthorized - Invalid or missing authentication token
422	Unprocessable Entity - Data validation failed

Common Use Cases

Use Case 1: Mobile App Network Monitoring

Mobile applications can use this endpoint to track network performance and connectivity patterns. When users experience connectivity issues, the app can log detailed network information including signal strength, network type, and location data to help identify problematic areas or network configurations.

Use Case 2: IoT Device Fleet Management

IoT devices deployed across multiple locations can regularly report their connectivity status, helping administrators monitor network coverage, identify devices experiencing poor connectivity, and optimize network infrastructure based on actual usage patterns and performance metrics.

Use Case 3: Enterprise Security Compliance

Organizations can track which networks their devices connect to, monitor security protocols in use, and ensure devices only connect to approved networks. This is particularly valuable for detecting unauthorized network access or identifying devices connecting to unsecured networks.

Use Case 4: Network Performance Analytics

Network administrators can collect aggregated connection data to analyze network performance trends, identify peak usage times, monitor data consumption patterns, and make informed decisions about network capacity and infrastructure improvements.

Use Case 5: Location-Based Services

Applications providing location-based services can correlate network connectivity data with geographic coordinates to understand connectivity patterns across different locations, optimize content delivery, and provide location-aware features based on network availability.

Best Practices

- **Data Validation**: Always validate network type and security values against the allowed enumeration before sending requests. Invalid values will result in validation errors.
- **Privacy Considerations**: When collecting location data (latitude/longitude), ensure compliance with privacy regulations and obtain appropriate user consent. Consider data anonymization for analytics purposes.
- **Batch Processing**: For applications generating high volumes of connection logs, consider implementing local queuing and batch processing to reduce API call frequency and improve performance.
- **Error Handling**: Implement robust error handling for network connectivity issues. Store logs locally when the API is unreachable and sync when connectivity is restored.
- **Data Completeness**: While many fields are optional, providing comprehensive data improves the value of analytics. Include quality metrics, data usage, and location information when available.
- **Security**: Never log sensitive authentication credentials or personal information. The API is designed to capture network characteristics, not user credentials or private data.
- **Rate Limiting**: Implement appropriate rate limiting in your applications to avoid overwhelming the API, especially for devices that frequently change networks or generate high-frequency connection events.

Revision #4
Created 2026-02-04 05:13:11 UTC by ipena@zequenze.com
Updated 2026-02-11 03:13:36 UTC by ipena@zequenze.com