

Device App Test Report

Endpoints Summary

Method	Path	Swagger
POST	/device_app_test_report/	Swagger ↗

“ The Device App Test Report API enables mobile and desktop applications to submit network performance test results to the GATE platform. This endpoint allows developers to register test data including download/upload speeds, latency, jitter, and DNS response times across different network types like WiFi, mobile, Bluetooth, and Ethernet.

Base URL: <https://gate.zequenze.com/api/v1>

Authentication: All endpoints require a Bearer token:

Authorization: Bearer <your-api-token>

Overview

The Device App Test Report API provides a centralized way to collect and store network performance metrics from client applications. This API is designed for applications that perform network diagnostics, speed tests, or connectivity monitoring and need to report their findings to a central analytics platform.

Key Features:

- Support for multiple network types (WiFi, Mobile, Bluetooth, Ethernet)
- Multiple test types including bandwidth, latency, jitter, and DNS performance
- Flexible destination targeting (URLs, hostnames, IP addresses)
- Standardized units and result formatting
- Batch test result reporting with count tracking

Common Integration Scenarios:

- Mobile apps performing periodic connectivity tests
- Network monitoring tools reporting performance metrics
- IoT devices submitting connectivity health data
- Desktop applications measuring network quality
- Quality assurance tools validating network performance

The API uses standardized abbreviations for network and test types to ensure consistent data collection across different client implementations.

Endpoints

POST /device_app_test_report/

Description: Submits network test results from client applications to the GATE platform. This endpoint accepts test data including network performance metrics, test metadata, and result values. Use this endpoint when your application has completed network tests and needs to report the results for analytics, monitoring, or troubleshooting purposes.

Use Cases:

- Mobile speed test apps reporting download/upload results
- Network monitoring tools submitting latency measurements
- IoT devices reporting connectivity health metrics
- Quality assurance applications validating network performance
- Diagnostic tools measuring DNS response times

Full URL Example:

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

Parameters:

Parameter	Type	In	Required	Description
data	string	body	Yes	JSON string containing the test report data with network type, test type, destination, and results

Request Body Schema:

Field	Type	Required	Description
network_name	string	No	Name of the current network (e.g., "Office WiFi", "Verizon LTE")
network_type	string	Yes	Network type code: 'wi' (WiFi), 'mo' (Mobile), 'bl' (Bluetooth), 'et' (Ethernet)
test_type	string	Yes	Test type code: 'dl' (Download), 'ul' (Upload), 'de' (Delay), 'jt' (Jitter), 'dn' (DNS response time)
destination	string	Yes	Target for the test: URL, hostname, or IP address
value	number	Yes	Numeric test result value
count	integer	No	Number of test iterations performed (useful for averaged results)
unit	string	No	Unit of measurement: 'Kbps' (Kilobits/sec), 'KBps' (Kilobytes/sec), 'ms' (Milliseconds)

cURL Example:

```
curl -X POST "https://gate.zequenze.com/api/v1/device_app_test_report/" \  
-H "Authorization: Bearer YOUR_API_TOKEN" \  
-H "Content-Type: application/json" \  
-d '{  
  "network_name": "Office WiFi",  
  "network_type": "wi",  
  "test_type": "dl",  
  "destination": "speedtest.example.com",  
  "value": 95.7,  
  "count": 3,  
  "unit": "Kbps"  
}'
```

Example Request Bodies:

Download Speed Test:

```
{
  "network_name": "Home WiFi 5GHz",
  "network_type": "wi",
  "test_type": "dl",
  "destination": "cdn.speedtest.net",
  "value": 150.5,
  "count": 5,
  "unit": "Kbps"
}
```

Mobile Network Latency Test:

```
{
  "network_name": "T-Mobile 5G",
  "network_type": "mo",
  "test_type": "de",
  "destination": "8.8.8.8",
  "value": 45.2,
  "count": 10,
  "unit": "ms"
}
```

DNS Response Time Test:

```
{
  "network_type": "et",
  "test_type": "dn",
  "destination": "google.com",
  "value": 12.8,
  "count": 1,
  "unit": "ms"
}
```

Example Response (201 Created):

```
{
  "id": 12847,
  "network_name": "Office WiFi",
  "network_type": "wi",
  "test_type": "dl",
  "destination": "speedtest.example.com",
}
```

```
"value": 95.7,  
"count": 3,  
"unit": "Kbps",  
"timestamp": "2024-01-15T14:32:18Z",  
"status": "processed"  
}
```

Response Codes:

Status	Description
201	Created - Test report successfully registered
400	Bad Request - Invalid data format or missing required fields
401	Unauthorized - Invalid or missing API token
422	Unprocessable Entity - Valid JSON but invalid field values
500	Internal Server Error - Server processing error

Common Use Cases

Use Case 1: Mobile Speed Testing Application

A mobile app that performs periodic network speed tests and reports results to monitor carrier performance across different locations and times.

```
# Report download speed test  
POST /device_app_test_report/ with mobile network download data
```

Use Case 2: IoT Device Connectivity Monitoring

IoT devices that need to report their network connectivity quality for remote monitoring and troubleshooting purposes.

```
# Report connectivity health metrics  
POST /device_app_test_report/ with latency and DNS response data
```

Use Case 3: Enterprise Network Quality Assurance

Corporate applications that validate network performance across office locations to ensure adequate connectivity for business applications.

```
# Report comprehensive network tests
POST /device_app_test_report/ with WiFi/Ethernet performance data
```

Use Case 4: Gaming Application Latency Monitoring

Gaming platforms that measure and report network latency to game servers for matchmaking and performance optimization.

```
# Report game server latency
POST /device_app_test_report/ with jitter and delay measurements
```

Use Case 5: Network Diagnostic Tools

Diagnostic applications that perform comprehensive network tests and need to centralize results for analysis and reporting.

```
# Report multi-test diagnostic results
POST /device_app_test_report/ with various test types and destinations
```

Best Practices

- **Standardize Network Names:** Use consistent naming conventions for network_name to enable better analytics (e.g., "Office_WiFi_5GHz" rather than varying formats)
- **Batch Related Tests:** When performing multiple test types on the same network, submit separate reports for each test type to maintain data clarity
- **Include Test Counts:** Always specify the count parameter when averaging multiple test runs to provide context for result reliability
- **Use Appropriate Units:** Match units to test types - use 'ms' for latency/jitter/DNS tests, 'Kbps' or 'KBps' for throughput tests
- **Handle Network Transitions:** Be aware that network_type may change during testing on mobile devices; submit separate reports when network switches occur

- **Error Handling:** Implement retry logic for failed submissions, especially on mobile networks where connectivity may be intermittent
 - **Data Validation:** Validate test results before submission to avoid reporting obviously invalid values (e.g., negative latency, impossibly high speeds)
 - **Privacy Considerations:** Be mindful of including sensitive information in network_name or destination fields, especially for enterprise deployments
-
-

Revision #4

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

Updated 2026-02-11 03:16:16 UTC by ipena@zequenze.com