

NordLayer MSP API Documentation.

Overview

The NordLayer Managed Service Provider (MSP) API allows MSP partners to manage organizations, plans, and usage reports for their customer base.

The API is REST-based, uses JSON for request and response payloads, and is available only over HTTPS.

Base URL:

```
https://partner-api.nordlayer.com/msp/v1
```

Authentication

Authentication method

The MSP API uses **API key authentication**.

Each request must include a valid API key issued to your MSP partner account. Requests without a valid API key are rejected.

API key format

API keys are opaque tokens with the following structure:

```
1 msp_<prefix>.<secret>
```

Example:

```
1 msp_ab12cdEFJJtZb0s7k1jH3fZVjh7X2mV3l2YtqK6s4
```

⚠ Notes

- The API key must be treated as a **secret**
- Do not share or expose the key in client-side code
- If a key is compromised, revoke it immediately and create a new one

Request headers

Include the API key in **one** of the following headers.

Preferred:

```
1 Authorization: ApiKey <api-key>
```

Alternative:

```
1 x-api-key: <api-key>
```

📘 Rules:

- **Exactly one** authentication header must be present
- Multiple values are **not allowed**
- Header names are case-insensitive

- Header values must match the expected format exactly

API Endpoints

Organizations Management

Create Organization

- Endpoint:** POST /organizations
- Description:** Creates a new organization under the MSP partner account
- Authentication:** API key

Request Body:

```

1 {
2   "title": "Organization Test",      // Required: 3-50 characters
3   "identifier": "org_test",        // Required: 3-40 characters, alphanumeric with underscores
4   "type": "standard",            // Required: organization type (standard, trial, etc.)
5   "is_suspendable": true,         // Optional: whether organization can be suspended (required
                                    // for trial type)
6   "email": "test@email.com",      // Optional: administrator email
7   "plan_identifier": "test_identifier" // Conditional: required for new packages
8 }
```

Response: 201 Created

```

1 {
2   "title": "Organization Test",
3   "identifier": "org_test",
4   "plan_identifier": "test_identifier"
5 }
```

List Organizations

- Endpoint:** GET /organizations
- Description:** Search and list organizations with filtering, ordering, and pagination support

Query Parameters:

- limit** : Number of results per page (max 200)
- offset** : Number of results to skip
- search** : Search by organization title or identifier
- filters[title]** : Filter by title
- filters[status]** : Filter by status
- order[createdAt]** : Sort by creation date (asc/desc)
- order[updatedAt]** : Sort by update date (asc/desc)
- order[identifier]** : Sort by identifier (asc/desc)

Response: 200 OK

```

1 [
2 {
3   "title": "Organization Test",
4   "identifier": "org_test",
5   "plan_identifier": "test_identifier"
6 }
```

Response Headers:

- **X-Total-Count** : Total number of organizations

Get Organization Details

- **Endpoint:** `GET /organizations/{identifier}`
- **Description:** Retrieves detailed information about a specific organization
- **Path Parameters:**
 - **identifier** : Organization identifier (string)

Response: 200 OK

```

1 {
2   "title": "Organization Test",
3   "identifier": "org_test",
4   "plan_identifier": "test_identifier"
5 }
```

Update Organization Title

- **Endpoint:** `PATCH /organizations/{identifier}`
- **Description:** Updates the title of an organization
- **Path Parameters:**
 - **identifier** : Organization identifier (string, 3-40 characters)

Request Body:

```

1 {
2   "name": "Organization's new title/name" // Required: 3-50 characters
3 }
```

Response: 200 OK

```

1 {
2   "title": "Organization's new title/name",
3   "identifier": "org_test",
4   "plan_identifier": "test_identifier"
5 }
```

Update Organization Status

- **Endpoint:** `PUT /organizations/{identifier}/status`
- **Description:** Updates the status of an organization
- **Path Parameters:**
 - **identifier** : Organization identifier (string, 3-40 characters)

Request Body:

```

1 {
2   "status": "suspended" // Required: one of "suspended", "subscribed", or "terminated"
3 }
```

Response: 200 OK

```

1 {
2   "title": "Organization Test",
```

```
3 "identifier": "org_test",
4 "plan_identifier": "test_identifier"
5 }
```

Plans Management

Get Plans

- **Endpoint:** GET /plans
- **Description:** Gets partner package and plans information

Response: 200 OK

```
1 [
2 {
3   "title": "some title",
4   "identifier": "some identifier",
5   "durationInterval": 2,
6   "durationUnit": "month",
7   "group": "advanced",
8   "items": [
9     {
10       "name": "Member",
11       "identifier": "member",
12       "price": 100.0,
13       "type": "member",
14       "group": "feature",
15       "config": {
16         "identifier": "fwass_10",
17         "tier": 10, // Optionally present
18         "setting": {
19           "price_unit": "organization",
20           "blocking_included": false
21         }
22       }
23     }
24   ]
25 }
26 ]
```

Usage Reports

Get Usage Reports

- **Endpoint:** GET /usage-reports
- **Description:** Get MSP usage reports with limit and offset pagination

Query Parameters:

- **limit** : Number of results per page (0-100, required)
- **offset** : Number of results to skip (>= 0, required)
- **organization_identifier** : Filter by organization identifier (optional)
- **date_from** : Start date in YYYY-MM-DD format (optional)
- **date_to** : End date in YYYY-MM-DD format (optional)

Response: 200 OK

```
1 [
2 {
3   "distributor_id": 1,
4   "partner_id": 2,
5   "partner_name": "Partner Name",
```

```
6  "organization_id": 3,  
7  "organization_name": "Organization Name",  
8  "license_type": "standard",  
9  "date": "2024-01-01",  
10 "amount": 100,  
11 "billable": 95,  
12 "organization_type": "standard",  
13 "plan_group": "Partner"  
14 }  
15 ]
```

Response Headers:

- **X-Total-Count** : Total number of usage reports

Error Handling

Common HTTP Status Codes

- **200 OK**: Request successful
- **201 Created**: Resource created successfully
- **400 Bad Request**: Invalid request data or validation errors
- **401 Unauthorized**: Authentication failed or missing
- **403 Forbidden**: Insufficient permissions
- **404 Not Found**: Resource not found
- **429 Too Many Requests**: Rate limiting
- **500 Internal Server Error**: Server error

Error Response Format

All error responses follow a consistent format:

```
1 {  
2   "message": "Error description",  
3   "code": 400  
4 }
```

Common Error Scenarios

Authentication Errors

```
1 {  
2   "message": "Authorization header not provided",  
3   "code": 401  
4 }
```

Validation Errors

```
1 {  
2   "message": "The email field is required",  
3   "code": 400  
4 }
```

Resource Not Found

```
1 {  
2   "message": "Organization not found",  
3   "code": 404  
4 }
```

Examples

Complete Organization Creation Workflow

```
1 API_KEY="msp_ab12cdEF.JJtZb0s7k1jH3fZVjhp7X2mV3l2YtqK6s4"
2
3 # 1. Create a new organization
4 curl -X POST "https://partner-api.nordlayer.com/msp/v1/organizations" \
5 -H "Authorization: ApiKey $API_KEY" \
6 -H "Content-Type: application/json" \
7 -d '{
8   "title": "New Organization",
9   "identifier": "new_org_2026",
10  "type": "standard",
11  "is_suspendable": true,
12  "plan_identifier": "standard_plan"
13 }'
14
15 # 2. Get the created organization details
16 curl -X GET "https://partner-api.nordlayer.com/msp/v1/organizations/new_org_2026" \
17 -H "Authorization: ApiKey $API_KEY"
18
```

Usage Report Retrieval

```
1 API_KEY="msp_ab12cdEF.JJtZb0s7k1jH3fZVjhp7X2mV3l2YtqK6s4"
2
3 # Get usage reports for a specific organization and date range
4 curl -X GET "https://partner-api.nordlayer.com/msp/v1/usage-reports?
organization_identifier=new_org_2026&date_from=2024-01-01&date_to=2024-01-
31&limit=100&offset=0" \
5 -H "Authorization: ApiKey $API_KEY"
6
7 # Get paginated usage reports
8 curl -X GET "https://partner-api.nordlayer.com/msp/v1/usage-reports?limit=100&offset=0" \
9 -H "Authorization: ApiKey $API_KEY"
```

Get Available Plans

```
1 API_KEY="msp_ab12cdEF.JJtZb0s7k1jH3fZVjhp7X2mV3l2YtqK6s4"
2
3 # Get available plans
4 curl -X GET "https://partner-api.nordlayer.com/msp/v1/plans" \
5 -H "Authorization: ApiKey $API_KEY"
```

Update Organization Title

```
1 API_KEY="msp_ab12cdEF.JJtZb0s7k1jH3fZVjhp7X2mV3l2YtqK6s4"
2
3 # Update organization title
4 curl -X PATCH "https://partner-api.nordlayer.com/msp/v1/organizations/new_org_2026" \
5 -H "Authorization: ApiKey $API_KEY" \
6 -H "Content-Type: application/json" \
7 -d '{
8   "name": "Updated Organization Name"
9 }'
```

Update Organization Status

```
1 API_KEY="msp_ab12cdEF.JJtZb0s7k1jH3fZVjhp7X2mV3l2YtqK6s4"
2
3 # Update organization status (subscribed, suspended, or terminated)
4 curl -X PUT "https://partner-api.nordlayer.com/msp/v1/organizations/new_org_2026/status" \
5 -H "Authorization: ApiKey $API_KEY" \
6 -H "Content-Type: application/json" \
7 -d '{
8   "status": "suspended"
9 }'
```

Authentication Error Scenarios

Invalid API Key

```

1  {
2  "message": "Invalid MSP Key",
3  "code": 401
4 }
```

Why this occurs: The authentication header was accepted, and the token format was valid, but the key could not be verified. This includes cases such as:

- The prefix does not correspond to an active key, or
- The secret does not match the key associated with the prefix, or
- The key exists but is not valid for verification (for example, not active)

How to fix:

- Verify you are using the correct API key and that it is not truncated.
- If the key may be compromised or out of sync, revoke it, create a new key, and update your integration.

Missing or malformed authentication header

```

1  {
2  "message": "Authorization header not provided",
3  "code": 401
4 }
```

Why this occurs: The request did not include a valid authentication header. This error is returned when:

- The `Authorization` header is missing, empty, or cannot be parsed, or
- The `Authorization` header is present but does not match the expected format `Authorization: ApiKey <token>`

How to fix:

- Send the header in the exact format:
 - `Authorization: ApiKey msp_<prefix>.<secret>`
- Ensure there is a single space after `ApiKey`.
- Ensure the token is complete (not truncated) and contains no leading or trailing whitespace.
- Ensure only one authentication header value is sent.