

Vertex Core API Reference Manual

High-Throughput Distributed Data Ledger & Sync Platform

VERTEX
v3.4.0-Stable

1 Introduction

Welcome to the **Vertex Core API Reference Manual**. The Vertex Core API provides developers with access to a high-throughput, low-latency distributed data ledger and replication network. Built on top of the robust Vertex Engine, this API enables secure multi-region synchronization, state monitoring, and automated transactional routing.

Whether you are integrating with our platform for financial reconciliation, high-frequency logistics synchronization, or decentralized identity networks, this reference manual outlines the specifications needed to construct correct and efficient integrations.

1.1 Base Environments

All requests to the Vertex API must be directed to one of the following host base URLs:

- **Production API Host:** <https://api.vertex.meridian.net/v2>
- **Sandbox API Host:** <https://sandbox-api.vertex.meridian.net/v2>

The sandbox host supports identical functionality to production, but runs on isolated test ledgers with simulated execution times and rate limits.

1.2 Rate Limits and Throttling

To guarantee service availability and prevent abuse, Vertex Core API enforces strict rate limits. Limits are tracked per client identity token and are reported via standard HTTP headers in every response:

- **X-RateLimit-Limit:** The maximum requests permitted per window.
- **X-RateLimit-Remaining:** The remaining request allocation in the current window.
- **X-RateLimit-Reset:** The Unix epoch timestamp when the current window resets.

Tier	Hourly Limit	Burst Capacity (rps)	Primary Use Case
Standard Sandbox	10,000	50	Application testing and prototyping
Production Team	50,000	200	Small to mid-scale enterprise workloads
Production Enterprise	500,000	1,500	High-frequency transaction sync

Table 1: Vertex Core API Rate Limit Allocation

When a limit is exceeded, the API returns an **HTTP 429 Too Many Requests** status code, accompanied by a JSON payload describing the backoff requirement (see Section 4).

2 Authentication

All requests to the Vertex API must be authenticated using JSON Web Tokens (JWT) [1] passed in the **Authorization** HTTP header. Requests without valid credentials will return an **HTTP 401 Unauthorized** response.

POST `/v2/auth/token`

AUTHENTICATION

Retrieve a short-lived JSON Web Token (JWT) using client credentials provided during developer onboarding.

To obtain a token, construct an **HTTP POST** request to the `/v2/auth/token` endpoint. The request body must contain your client credentials encoded as JSON.

2.0.1 Request Parameters

- `client_id` (*string, required*): Your organization's unique identifier.
- `client_secret` (*string, required*): Your API client secret key.
- `audience` (*string, optional*): Target environment identifier. Defaults to `vertex-ledger`.

Example Request (JSON)

```
{
  "client_id": "nex_98a72f",
  "client_secret": "vtx_sec_77218ac89e",
  "audience": "vertex-ledger-prod"
}
```

Example Response (200 OK)

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZWQ...",
  "expires_in": 3600,
  "token_type": "Bearer",
  "scope": "read:jobs write:jobs"
}
```

2.1 Using the Token

Once retrieved, include the token in all subsequent requests within the `Authorization` header as a Bearer token:

```
// Example HTTP Header
Authorization: Bearer eyJhbGciOiJIUzI1NiIsIn...
```

3 Data Synchronization Endpoints

The Synchronization API provides tools for managing multi-region sync operations across your Vertex datastores.

GET `/v2/sync/jobs`

DATA SYNC

Retrieve a paginated list of all active synchronization jobs associated with your account.

3.0.1 Query Parameters

- **status** (*string, optional*): Filter by status (`pending`, `running`, `completed`, `failed`).
- **limit** (*integer, optional*): Number of records to return. Default is 20. Max is 100.

- `cursor` (*string, optional*): Pagination cursor for the next page of results.

Request Headers

```
GET /v2/sync/jobs?status=running&limit=1
HTTP/1.1
Host: api.vertex.meridian.net
Authorization: Bearer eyJhbGc...
```

Response Payload

```
{
  "data": [
    {
      "job_id": "job_vtx_00192a7",
      "status": "running",
      "source_region": "us-east-1",
      "target_region": "eu-west-1",
      "records_synced": 482910,
      "created_at": "2026-08-01T12:00:00Z"
    }
  ],
  "pagination": {
    "next_cursor": "c3luY19qb2JfMDAxOTJhNw",
    "has_more": true
  }
}
```

POST `/v2/sync/jobs`

DATA SYNC

Initialize a new real-time synchronization pipeline between source and target clusters.

3.0.2 Request Body Parameters

- `source_id` (*string, required*): Fictional cluster ID containing primary tables.
- `target_id` (*string, required*): Target cluster ID to receive replicas.
- `sync_mode` (*string, required*): The replication level (`snapshot`, `realtime`, or `differential`).
- `filters` (*array, optional*): Specific databases/tables to replicate.

Example Request (JSON)

```
{
  "source_id": "clus_nexa_9910",
  "target_id": "clus_apex_4201",
  "sync_mode": "realtime",
  "filters": ["sales_data", "customers"]
}
```

Example Response (201 Created)

```
{
  "job_id": "job_vtx_00192b8",
  "status": "pending",
  "source_id": "clus_nexa_9910",
  "target_id": "clus_apex_4201",
  "sync_mode": "realtime",
  "estimated_bytes": 10737418240,
  "created_at": "2026-08-03T23:25:00Z"
}
```

DELETE

/v2/sync/jobs/{id}

DATA SYNC

Gracefully terminate an active synchronization pipeline. This will stop replication but will not delete synced tables.

3.0.3 Path Parameters

- **id** (*string, required*): The unique job identifier, prefixed with `job_vtx_`.

Example Request

```
DELETE /v2/sync/jobs/job_vtx_00192b8
HTTP/1.1
Host: api.vertex.meridian.net
Authorization: Bearer eyJhbGc...
```

Example Response (200 OK)

```
{
  "job_id": "job_vtx_00192b8",
  "status": "terminated",
  "terminated_at": "2026-08-03T23:27:00Z",
  "final_sync_bytes": 4518290
}
```

4 Error Handling

The Vertex Core API returns standard HTTP response codes to indicate the success or failure of an API request. In general:

- Codes in the **2xx** range indicate success.
- Codes in the **4xx** range indicate an error that failed because of the information provided (e.g., a missing parameter, authentication failure, etc.).
- Codes in the **5xx** range indicate an error on the Vertex server infrastructure.

4.1 Error Response Body

When a request fails, additional diagnostics are returned in the response body as a JSON object containing the error details:

```
{
  "error": {
    "code": "INVALID_PARAMETER",
    "message": "The request body parameter 'sync_mode' is invalid.",
    "doc_url": "https://docs.vertex.meridian.net/errors/INVALID_PARAMETER",
    "details": [
      {
        "field": "sync_mode",
        "issue": "Must be one of: snapshot, realtime, differential"
      }
    ]
  }
}
```

4.2 Error Codes and Types

Table 2 lists the common API-specific errors that you may encounter while integrating with the Vertex Core API.

API Error Code	HTTP Status	Description
UNAUTHORIZED	401 Unauthorized	The access token provided is invalid or has expired. Obtain a new token using /v2/auth/token .
FORBIDDEN	403 Forbidden	Your organization lacks the permissions required to access the requested resource.
RESOURCE_NOT_FOUND	404 Not Found	The requested resource (e.g. cluster ID or job ID) does not exist or belongs to another tenant.
INVALID_PARAMETER	400 Bad Request	One or more request parameters (query, header, or body) failed to meet validation requirements.
RATE_LIMIT_EXCEEDED	429 Too Many Requests	The client has exceeded its rate limit. Backoff is required before sending more requests.
LEDGER_LOCKED	409 Conflict	The operation could not be completed because the target replication ledger is currently locked for maintenance.
INTERNAL_ERROR	500 Server Error	An internal error occurred in the Vertex replication service. Retry after standard backoff intervals.

Table 2: Vertex Core API Common Error Codes

References

- [1] M. Jones, J. Bradley, and N. Sakimura. Json web token (jwt). RFC 7519, IETF, May 2015.