DataFlow API - Technical Reference

Introduction

The DataFlow API is a RESTful web service that enables real-time data integration and transformation. It supports JSON and XML formats with webhook notifications for asynchronous processing. This manual covers API v2.0 released in October 2024. Base URL: https://api.dataflow.io/v2

Authentication

All API requests require authentication using API keys. Include your key in the Authorization header: *Authorization: Bearer YOUR_API_KEY*. Generate API keys from the dashboard under Settings > API Keys. Keys can have read-only or read-write permissions. Rotate keys every 90 days for security. Rate limits: 1000 requests per hour for standard plans, 10000 for enterprise.

Endpoints - Data Ingestion

POST /api/v2/ingest
Upload data for processing. Accepts JSON payload up to 10MB.
Request body: {"source": "string", "data": "object", "transformations": "array"}
Response: {"job_id": "string", "status": "queued", "eta_seconds": integer}
Example: POST https://api.dataflow.io/v2/ingest with Content-Type: application/json

Endpoints - Job Status

GET /api/v2/jobs/{job_id}
Check processing status of a submitted job.
Path parameter: job_id (string, required)
Response: {"job_id": "string", "status": "processing|completed|failed", "progress": integer, "result_url": "string"}
Status values: queued, processing, completed, failed. Poll every 5-10 seconds.

Endpoints - Data Retrieval

GET /api/v2/data/{dataset_id}
Retrieve processed data by dataset ID.
Query parameters: format (json|xml|csv), limit (integer, max 1000), offset (integer)
Response: {"dataset_id": "string", "records": array, "total_count": integer, "next_page": "string"}
Use pagination for large datasets. Cache responses for 5 minutes.

Data Transformations

The API supports built-in transformations applied during ingestion. Available transformations: filter (remove records by condition), map (transform field values), aggregate (group and summarize), join (merge with reference data), validate (schema validation). Specify transformations as an array in the ingest payload. Transformations are applied in order. Complex transformations may increase processing time.

Error Handling

API returns standard HTTP status codes. 200: Success, 400: Bad Request (validation error), 401: Unauthorized (invalid API key), 429: Rate Limit Exceeded, 500: Internal Server Error. Error responses include: {"error": "string", "message": "string", "code": "string"}. Retry failed requests with exponential backoff. Contact support if errors persist for 500 status codes.

Webhooks

Configure webhooks to receive notifications when jobs complete. Set webhook URL in dashboard under Settings > Webhooks. Webhook payload: {"event": "job.completed", "job_id": "string", "status": "string", "timestamp": "ISO8601"}. Webhooks include HMAC signature in X-Signature header for verification. Respond with 200 status within 5 seconds. Failed webhook deliveries are retried 3 times.

Best Practices

1. Batch multiple records in a single ingest request to reduce API calls.
2. Use webhooks instead of polling for job status when possible.
3. Implement exponential backoff for rate limit errors.
4. Cache GET responses to minimize redundant requests.
5. Validate data client-side before sending to reduce processing failures.

6. Use compression (gzip) for large payloads to improve transfer speed.
7. Monitor your API usage in the dashboard to avoid unexpected overages.

SDK Libraries

Official SDK libraries are available for Python, JavaScript, Java, and Go. Install via package managers: pip install dataflow-sdk, npm install @dataflow/sdk. SDKs handle authentication, retries, and pagination automatically. Refer to SDK documentation at docs.dataflow.io/sdks. Community libraries exist for Ruby, PHP, and .NET. Example code snippets are provided in the documentation for common use cases.