DataSync API Reference

Version 2.1.0 | Last Updated: February 2026

1. Introduction

Welcome to the DataSync API documentation. DataSync is a powerful, RESTful API that enables real-time data synchronization across distributed systems. This documentation provides comprehensive guidance for integrating DataSync into your applications and managing data workflows at scale.

DataSync supports multiple data formats, provides robust error handling, and includes built-in retry mechanisms. With support for webhooks and real-time notifications, you can build responsive applications that stay synchronized with your data sources.

> **Getting Started**
>
> New to DataSync? Start with the Installation section to set up your first connection, then review the Configuration section to customize your sync behavior.

2. Installation

Installing DataSync is straightforward and supports multiple deployment environments including Docker, Kubernetes, and traditional server setups.

Docker Installation

The easiest way to get started is using the official Docker image:

```
docker pull datasync:latest
docker run -d \
  -p 8080:8080 \
  -e API_KEY=your-api-key \
  -e DB_URL=postgresql://user:pass@db:5432/sync \
  datasync:latest
```

Kubernetes Deployment

For Kubernetes environments, use the official Helm chart:

```
helm repo add datasync https://charts.datasync.io
helm install datasync datasync/datasync \
  --set apiKey=your-api-key \
  --set database.url=postgresql://db:5432/sync
```

> **Security Warning**
>
> Never commit API keys to version control. Use environment variables or secrets management systems like HashiCorp Vault, AWS Secrets Manager, or Kubernetes Secrets.

3. Configuration

DataSync configuration is managed through environment variables and YAML configuration files. The system supports hot-reload of non-critical settings without service interruption.

Environment Variables

Key configuration environment variables:

- `API_KEY` - Authentication token for API access (required)
- `DB_URL` - Database connection string (required)
- `SYNC_INTERVAL` - Sync check interval in seconds (default: 30)
- `MAX_RETRIES` - Maximum retry attempts (default: 3)
- `WEBHOOK_TIMEOUT` - Webhook request timeout in seconds (default: 30)
- `LOG_LEVEL` - Logging level: DEBUG, INFO, WARN, ERROR (default: INFO)

YAML Configuration

For advanced configuration, create a `datasync.yaml` file:

```yaml
server:
  port: 8080
  timeout: 30s

database:
  url: postgresql://localhost/datasync
  pool:
    min: 5
    max: 20

sync:
  interval: 30s
  maxRetries: 3
  backoff: exponential

webhook:
  timeout: 30s
  retries: 3
```

4. API Reference

The DataSync API provides endpoints for managing synchronization tasks, monitoring status, and retrieving sync results. All requests require authentication via API key in the `Authorization` header.

Authentication

Include your API key in the `Authorization` header for all requests:

```
Authorization: Bearer your_api_key_here
```

Endpoints

Method	Path	Description	Response
POST	/api/v1/sync	Create a new synchronization task	201 Created with sync ID
GET	/api/v1/sync/{id}	Retrieve sync task details and status	200 OK with task object
GET	/api/v1/sync/{id}/status	Get real-time sync status updates	200 OK with status object
PATCH	/api/v1/sync/{id}	Update synchronization configuration	200 OK with updated task
DELETE	/api/v1/sync/{id}	Cancel an active synchronization task	204 No Content

Example Request/Response

Create Sync Task

Request:

```
POST /api/v1/sync HTTP/1.1
Host: api.datasync.io
Authorization: Bearer your_api_key
Content-Type: application/json

{
  "source": {
    "type": "postgresql",
    "url": "postgresql://source-db:5432/mydb"
  },
  "destination": {
    "type": "postgresql",
    "url": "postgresql://dest-db:5432/mydb"
  },
  "tables": ["users", "transactions"],
  "syncInterval": "30s"
}
```

Response (201 Created):

```
{
  "id": "sync_abc123xyz789",
  "status": "running",
  "progress": 0,
  "createdAt": "2026-02-21T10:30:00Z",
  "source": {...},
  "destination": {...},
  "tables": ["users", "transactions"],
  "estimatedDuration": "2m45s"
}
```

> **API Rate Limits**
>
> DataSync API enforces rate limits of 1000 requests per minute per API key. Responses include `X-RateLimit-Remaining` and `X-RateLimit-`

> `Reset` headers to help you manage your quota.

5. Troubleshooting

Common issues and their solutions:

Connection Timeout

If sync tasks timeout, increase the `WEBHOOK_TIMEOUT` environment variable and check network connectivity between services. Ensure firewall rules allow communication on required ports.

Authentication Errors

Verify your API key is correct and hasn't expired. Check that the key is included in the `Authorization` header with the format `Bearer <key>`.

Database Connection Issues

Test database connectivity using standard database clients. Verify credentials, hostname, and port in the connection string. Check that the database user has appropriate permissions for read/write operations.

> **Performance Considerations**
>
> For large datasets exceeding 1GB, consider implementing incremental sync strategies and enabling compression. Monitor resource usage during sync operations to prevent service degradation.