THANOS API Reference

Overview

The THANOS system provides RESTful APIs for file organization, classification, and management. This document covers all available endpoints and their usage.

Base URL

```
Development: http://localhost:3000/api
Production: https://your-domain.com/api
```

Authentication

All API endpoints require authentication via API key:

```
Authorization: Bearer YOUR_API_KEY
Content-Type: application/json
```

Endpoints

1. File Upload

Upload files for organization.

```
POST /api/upload
```

Request Body:

```
{
  "files": ["file1", "file2"],
  "metadata": {
    "user_id": "string",
    "tier": "Standard|Pro|Veteran"
  }
}
```

Response:

```
{
  "success": true,
  "uploaded_files": [
    {
      "id": "file_id",
      "name": "filename.jpg",
      "size": 1024000,
      "path": "/uploads/filename.jpg"
    }
  ],
  "total_size": 1024000
}
```

2. THANOS Organize

Trigger the main organization process.

```
POST /api/organize
```

Request Body:

```
{
  "job_id": "uuid",
  "scope": "/path/to/files",
  "tier": "Standard|Pro|Veteran",
  "dry_run": false,
  "preferences": {
    "create_date_folders": true,
    "create_people_folders": true,
    "create_type_folders": true
  }
}
```

Response:

```
{
  "job_id": "uuid",
  "status": "processing",
  "estimated_time": 120,
  "files_found": 150,
  "progress_url": "/api/progress/job_id"
}
```

3. Organization Status

Get real-time status of organization job.

```
GET /api/organize/{job_id}/status
```

Response:

```
{
  "job_id": "uuid",
  "status": "processing|completed|failed",
  "progress": {
    "current": 75,
    "total": 150,
    "percentage": 50
  },
  "current_stage": "classification",
  "time_elapsed": 60,
  "estimated_remaining": 60,
  "files_processed": 75,
  "errors": []
}
```

4. Organization Results

Get detailed results of completed organization.

```
GET /api/organize/{job_id}/results
```

Response:

```
{
  "job_id": "uuid",
  "summary": {
    "total_files": 150,
    "files_organized": 145,
    "folders_created": 12,
    "files_skipped": 5,
    "processing_time": 180
  },
  "folder_structure": [
    {
      "path": "Photos/2023/08-August",
      "file_count": 45,
      "total_size": 15728640
    }
  ],
  "file_movements": [
    {
      "original_path": "/unorganized/IMG_001.jpg",
      "new_path": "/Photos/2023/08-August/IMG_001.jpg",
      "classification": "photo",
      "confidence": 0.95
    }
  ],
  "undo_token": "undo_token_string"
}
```

5. Undo Organization

Revert an organization operation.

```
POST /api/undo
```

Request Body:

```
{
  "job_id": "uuid",
  "undo_token": "string"
}
```

Response:

```
{
  "success": true,
  "files_reverted": 145,
  "folders_removed": 12,
  "message": "Organization successfully reverted"
}
```

6. File Statistics

Get file statistics and insights.

```
GET /api/stats
```

Query Parameters:
- `scope` : File scope (optional)
- `period` : Time period (day|week|month|year)

Response:

```
{
  "total_files": 1250,
  "total_size": 5368709120,
  "by_category": {
    "photos": {
      "count": 800,
      "size": 3221225472,
      "percentage": 64
    },
    "documents": {
      "count": 350,
      "size": 1073741824,
      "percentage": 28
    }
  },
  "recent_activity": [
    {
      "date": "2023-08-15",
      "files_organized": 150,
      "time_saved": "2 hours"
    }
  ],
  "organization_efficiency": 0.92
}
```

7. Organization History

Get history of organization jobs.

```
GET /api/organizations
```

Query Parameters:

- `limit`: Number of results (default: 20)
- `offset`: Pagination offset (default: 0)
- `status`: Filter by status (optional)

Response:

```
{
  "organizations": [
    {
      "job_id": "uuid",
      "created_at": "2023-08-15T10:30:00Z",
      "completed_at": "2023-08-15T10:33:00Z",
      "status": "completed",
      "files_processed": 150,
      "time_taken": 180,
      "tier": "Pro"
    }
  ],
  "total": 45,
  "has_more": true
}
```

Webhooks

Configure webhooks to receive real-time updates about organization progress.

Webhook Events

- `organization.started`
- `organization.progress`
- `organization.completed`
- `organization.failed`

Webhook Payload

```
{
  "event": "organization.progress",
  "job_id": "uuid",
  "timestamp": "2023-08-15T10:30:00Z",
  "data": {
    "progress": 50,
    "current_stage": "classification",
    "files_processed": 75
  }
}
```

Error Responses

All endpoints return consistent error responses:

```json
{
  "error": {
    "code": "ERROR_CODE",
    "message": "Human readable error message",
    "details": "Additional error details",
    "timestamp": "2023-08-15T10:30:00Z"
  }
}
```

Common Error Codes

- `INVALID_REQUEST` : Request validation failed
- `UNAUTHORIZED` : Authentication failed
- `FORBIDDEN` : Insufficient permissions
- `NOT_FOUND` : Resource not found
- `RATE_LIMITED` : Too many requests
- `QUOTA_EXCEEDED` : User quota exceeded
- `INTERNAL_ERROR` : Server error

Rate Limits

API rate limits are tier-based:

Tier	Requests/Hour	Files/Hour
Standard	100	500
Pro	1000	5000
Veteran	10000	50000

Rate limit information is included in response headers:

```
X-RateLimit-Limit: 1000
X-RateLimit-Remaining: 999
X-RateLimit-Reset: 1692097200
```

SDK Examples

JavaScript/Node.js

```javascript
const ThanosClient = require('@thanos/client');

const client = new ThanosClient({
  apiKey: 'your-api-key',
  baseUrl: 'https://api.thanos.example.com'
});

// Upload files
const uploadResult = await client.upload({
  files: ['file1.jpg', 'file2.pdf'],
  metadata: { tier: 'Pro' }
});

// Organize files
const organizeResult = await client.organize({
  scope: '/uploads',
  tier: 'Pro',
  dry_run: false
});

// Monitor progress
client.onProgress(organizeResult.job_id, (progress) => {
  console.log(`Progress: ${progress.percentage}%`);
});
```

Python

```python
from thanos_client import ThanosClient

client = ThanosClient(
    api_key='your-api-key',
    base_url='https://api.thanos.example.com'
)

# Upload files
upload_result = client.upload(
    files=['file1.jpg', 'file2.pdf'],
    metadata={'tier': 'Pro'}
)

# Organize files
organize_result = client.organize(
    scope='/uploads',
    tier='Pro',
    dry_run=False
)

# Get results
results = client.get_results(organize_result['job_id'])
```

cURL Examples

```
# Upload files
curl -X POST https://api.thanos.example.com/api/upload \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -F "files=@photo1.jpg" \
  -F "files=@document.pdf"

# Organize files
curl -X POST https://api.thanos.example.com/api/organize \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "scope": "/uploads",
    "tier": "Pro",
    "dry_run": false
  }'

# Check status
curl -X GET https://api.thanos.example.com/api/organize/job-id/status \
  -H "Authorization: Bearer YOUR_API_KEY"
```

Testing

Use the sandbox environment for testing:

```
Sandbox URL: https://sandbox-api.thanos.example.com
Test API Key: test_12345
```

The sandbox environment includes:
- Mock file processing (fast responses)
- Sample file datasets
- All API endpoints
- Webhook testing support

For additional support, see the FAQ (FAQ.md) or contact support.