# DocHub Pipeline Test Dashboard User Guide

## Introduction

The DocHub Pipeline Test Dashboard is a powerful diagnostic and visualization tool designed to help developers and advanced users monitor, debug, and understand the document processing pipeline. The dashboard provides a comprehensive view of each processing stage, from text extraction to knowledge graph generation, making it an invaluable tool for troubleshooting issues and optimizing pipeline performance.

## Table of Contents

1. [Dashboard Access](#dashboard-access)
2. [User Interface Overview](#user-interface-overview)
3. [Document Selection](#document-selection)
4. [Pipeline Steps Tab](#pipeline-steps-tab)
5. [Text Chunks Tab](#text-chunks-tab)
6. [Entities Tab](#entities-tab)
7. [Relationships Tab](#relationships-tab)
8. [Graph Preview Tab](#graph-preview-tab)
9. [Error Debug Tab](#error-debug-tab)
10. [Developer API Reference](#developer-api-reference)

## Dashboard Access

To access the Pipeline Test Dashboard:

1. Process a document through the system
2. Note the document’s UUID
3. Navigate to: http://your-server/dochub/test-dashboard/<document-uuid>/

Alternatively, you can access the dashboard from the document detail page by selecting “View Processing Details” from the actions menu.

## User Interface Overview

The dashboard features a modern, dark-themed interface with several key components:

* **Header Area**: Contains the dashboard title and description
* **Document Selection Panel**: For selecting documents to analyze
* **Tab Navigation**: Provides access to different views of the processing data
* **Content Area**: Displays the selected tab’s content

The dashboard is organized into six main tabs: - Pipeline Steps - Text Chunks - Entities - Relationships - Graph Preview - Error Debug

## Document Selection

The Document Selection panel allows you to choose which document to analyze:

1. Click the “Load Documents” button to populate the dropdown list
2. Select a document from the dropdown menu
3. The dashboard will automatically load processing data for the selected document
4. Use the “Refresh Logs” button to update the dashboard with the latest processing information

### Document Status Indicators

Each document in the dropdown list shows its current status: - **Processing**: The document is currently being processed - **Ready**: The document has been successfully processed - **Error**: An error occurred during processing

## Pipeline Steps Tab

The Pipeline Steps tab provides a chronological timeline of the document’s journey through the processing pipeline.

### Key Features

* **Timeline View**: Visualizes the sequence of processing stages
* **Stage Cards**: Each major stage (extraction, splitting, embedding, etc.) is displayed as a card
* **Status Indicators**: Color-coded indicators show the status of each step:
  + Green: Completed successfully
  + Blue: Started
  + Yellow: In progress
  + Red: Error occurred
* **Timestamps**: Each log entry includes a timestamp for chronological tracking
* **Detailed Information**: Click “Details” on any log entry to view the complete processing data

### How to Use

1. Examine the timeline to understand which stages have completed and which may have issues
2. Expand the “Details” section on any log entry to see the raw processing data
3. Check timestamps to identify potential bottlenecks in the pipeline

## Text Chunks Tab

The Text Chunks tab displays the document text after it has been split into manageable chunks for processing.

### Key Features

* **Success Banner**: Shows confirmation when chunks are successfully extracted
* **Chunk Count**: Displays the total number of chunks created from the document
* **Individual Chunks**: Each chunk is displayed separately with its index number
* **Content Preview**: Shows the actual text content of each chunk

### How to Use

1. Verify that the document has been properly chunked
2. Check the chunk sizes to ensure they’re appropriate for your use case
3. Review chunk content to verify that the splitting logic preserved meaning
4. Use this information to debug issues with entity extraction or embedding quality

## Entities Tab

The Entities tab displays all entities extracted from the document during the knowledge graph creation process.

### Key Features

* **Tabular View**: Entities are displayed in a structured table
* **Entity Types**: Each entity is color-coded by its type (Person, Organization, Location, etc.)
* **Source Information**: Shows which chunk the entity was extracted from
* **Property Inspector**: Click “View Properties” to see all metadata for each entity

### How to Use

1. Review extracted entities to verify accuracy and completeness
2. Check the entity types to ensure proper classification
3. Examine entity properties for additional contextual information
4. Identify potential issues with entity extraction or classification

## Relationships Tab

The Relationships tab shows the connections between entities extracted from the document.

### Key Features

* **Relationship Table**: Displays source entity, relationship type, and target entity
* **Color Coding**: Source and target entities are color-coded by entity type
* **Chunk Reference**: Shows which chunk the relationship was extracted from
* **Property Details**: Click “View Properties” to see additional relationship metadata

### How to Use

1. Examine entity relationships to verify logical connections
2. Check relationship types for accuracy
3. Analyze which chunks yielded the most relationship data
4. Identify potential issues with relationship extraction

## Graph Preview Tab

The Graph Preview tab provides an interactive visualization of the knowledge graph created from the document.

### Key Features

* **Interactive Graph**: Visual representation of entities and their relationships
* **Navigation Controls**: Zoom in/out, fit view, and pan around the graph
* **Node Selection**: Click on nodes to view detailed entity information
* **Color Coding**: Entities are color-coded by type for easy identification
* **Graph Legend**: Explains the color coding of different entity types
* **Node Details Panel**: Shows detailed information about selected entities

### How to Use

1. Navigate the graph using the control buttons (Zoom In, Zoom Out, Fit View)
2. Click on any node to view detailed information about that entity
3. Examine the connections between entities to understand document relationships
4. Use the “View Full Graph” button for a larger view
5. Click “Refresh Graph” to update the visualization with latest data

## Error Debug Tab

The Error Debug tab provides detailed information for troubleshooting processing issues.

### Key Features

* **Document Status Card**: Shows the current processing status of the document
* **Component Status Cards**: Displays the status of each pipeline component
* **Error Details**: Shows specific error messages and stack traces when available
* **Troubleshooting Tips**: Provides guidance for resolving common issues

### How to Use

1. Check the document status to confirm the overall processing status
2. Review each component’s status to identify where issues may have occurred
3. Examine error details for specific error messages and stack traces
4. Refer to the troubleshooting tips for guidance on resolving issues
5. Use the detailed information to fix problems in your implementation

## Developer API Reference

The Pipeline Test Dashboard communicates with the backend through several API endpoints. This section documents these endpoints for developers who want to integrate with or extend the dashboard functionality.

### Document API Endpoints

#### Get Document Logs

GET /api/dochub/documents/{document\_id}/logs/

Returns the processing logs for a specific document, including details for each pipeline stage.

#### Get Document Chunks

GET /api/dochub/documents/{document\_id}/chunks/

Returns the text chunks created during document processing.

#### Get Document Status

GET /api/dochub/documents/status/{document\_id}/

Returns the current processing status of a document.

### Graph API Endpoints

#### Get Document Graph

GET /api/dochub/graph/document/{document\_id}/

Returns the knowledge graph data for a document, including nodes (entities) and links (relationships).

#### Get Entity Graph

GET /api/dochub/graph/entity/?name={entity\_name}&type={entity\_type}

Returns graph data for a specific entity, showing its relationships across documents.

### Data Structures

#### Pipeline Logs Structure

{  
 "document\_id": "uuid",  
 "document\_name": "example.pdf",  
 "status": "ready|processing|error",  
 "logs": [  
 {  
 "stage": "text\_extraction|text\_splitting|embedding\_generation|vector\_indexing|graph\_extraction|neo4j\_store\_graph",  
 "status": "started|completed|error|in\_progress",  
 "timestamp": "2025-03-15T12:34:56Z",  
 "details": { /\* Stage-specific details \*/ }  
 }  
 ]  
}

#### Graph Data Structure

{  
 "nodes": [  
 {  
 "id": "node\_id",  
 "name": "Entity Name",  
 "group": "Person|Organization|Location|etc.",  
 "properties": { /\* Entity properties \*/ }  
 }  
 ],  
 "links": [  
 {  
 "source": "source\_node\_id",  
 "target": "target\_node\_id",  
 "type": "relationship\_type",  
 "properties": { /\* Relationship properties \*/ }  
 }  
 ]  
}

### JavaScript Interface

The dashboard exposes several JavaScript functions that can be useful for extension or customization:

* loadDocumentLogs(documentId): Loads processing logs for a document
* renderTimeline(logs): Renders the processing timeline
* renderChunks(logs): Renders document text chunks
* renderEntities(nodes): Renders entity table
* renderRelationships(links, nodes): Renders relationship table
* renderGraphVisualization(graphData): Renders the knowledge graph visualization
* updateErrorDebugging(data): Updates the error debugging section

## Conclusion

The DocHub Pipeline Test Dashboard is an essential tool for monitoring, debugging, and understanding the document processing pipeline. By providing detailed visibility into each stage of processing, it helps developers optimize performance, identify issues, and ensure the quality of document processing within the DocHub system.