📖 PDF Ingestion & Search API Documentation

This document provides comprehensive instructions on how to use the PDF Ingestion & Search API, built with Node.js, Express, MongoDB, and Meilisearch. It covers authentication, PDF ingestion, and search functionalities, with example requests, responses, and setup instructions for both the deployed service on Render and local development. Screenshots (to be provided) will illustrate API usage in tools like Postman.

📋 Overview

The API is built with Node.js, Express, MongoDB, and Meilisearch, and is secured with JWT authentication. It is deployed on Render and can be tested directly using the deployed endpoints. Below, you'll find details on the key endpoints, example data, and instructions for configuring Meilisearch (either using the deployed instance or setting it up locally).

🔍 API Endpoints

1. POST /ingest

Description: Uploads and stores parsed PDF data in MongoDB and indexes it in Meilisearch for searchability.

Authentication: Requires a valid JWT token in the Authorization header.

Request:

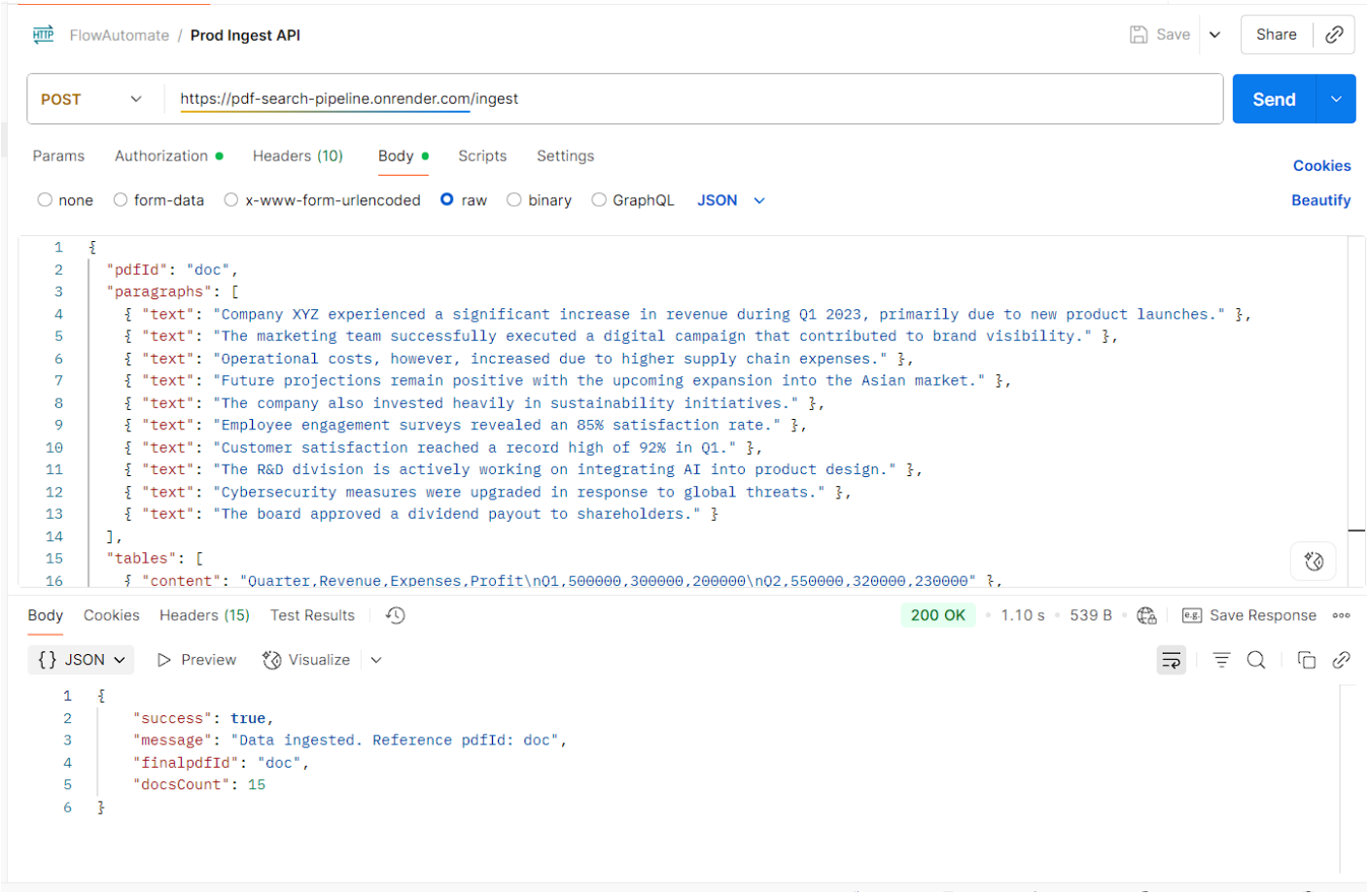
- **Method**: POST
- **Local URL**: http://localhost:5000/ingest
- **URL**: https://pdf-search-pipeline.onrender.com/ingest
- **Headers**: Content-Type: application/json
- **Authorization**: Bearer Token: F121523A
- **Body** (JSON): raw

```
{
  "pdfId": "doc",
  "paragraphs": [
    { "text": "Company XYZ experienced a significant increase in revenue during Q1 2023, primarily due to new product launches." },
    { "text": "The marketing team successfully executed a digital campaign that contributed to brand visibility." }],
  "tables": [
    { "content": "Quarter,Revenue,Expenses,Profit\nQ1,500000,300000,200000\nQ2,550000,320000,230000" },
    { "content": "Department,Employees\nEngineering,120\nMarketing,45\nSales,60" },
    { "content": "Region,Revenue\nNorth America,250000\nEurope,150000\nAsia,100000" }
  ],
  "images": [
    { "caption": "Figure 1: Revenue growth by quarter." },
    { "caption": "Figure 2: Regional revenue distribution." }
  ]
}
```

- **Response** (Success):

```
{
    "success": true,
    "message": "Data ingested. Reference pdfId: doc",
    "finalpdfId": "doc",
    "docsCount": 15
}
```


- **Screenshot**:



---

2. POST /search

Description: Searches for documents using Meilisearch based on a query string.

Authentication: Requires a valid JWT token in the Authorization header.

Request:

- **Method**: POST
- **Local URL**: http://localhost:5000/search
- **URL**: https://pdf-search-pipeline.onrender.com/search
- **Meili Search URL:** https://meili--meilisearch--6y7qslcbcmz5.code.run/
- **Headers**: Content-Type: application/json
- **Authorization**: Bearer Token: F121523A
- **Body** (JSON): raw

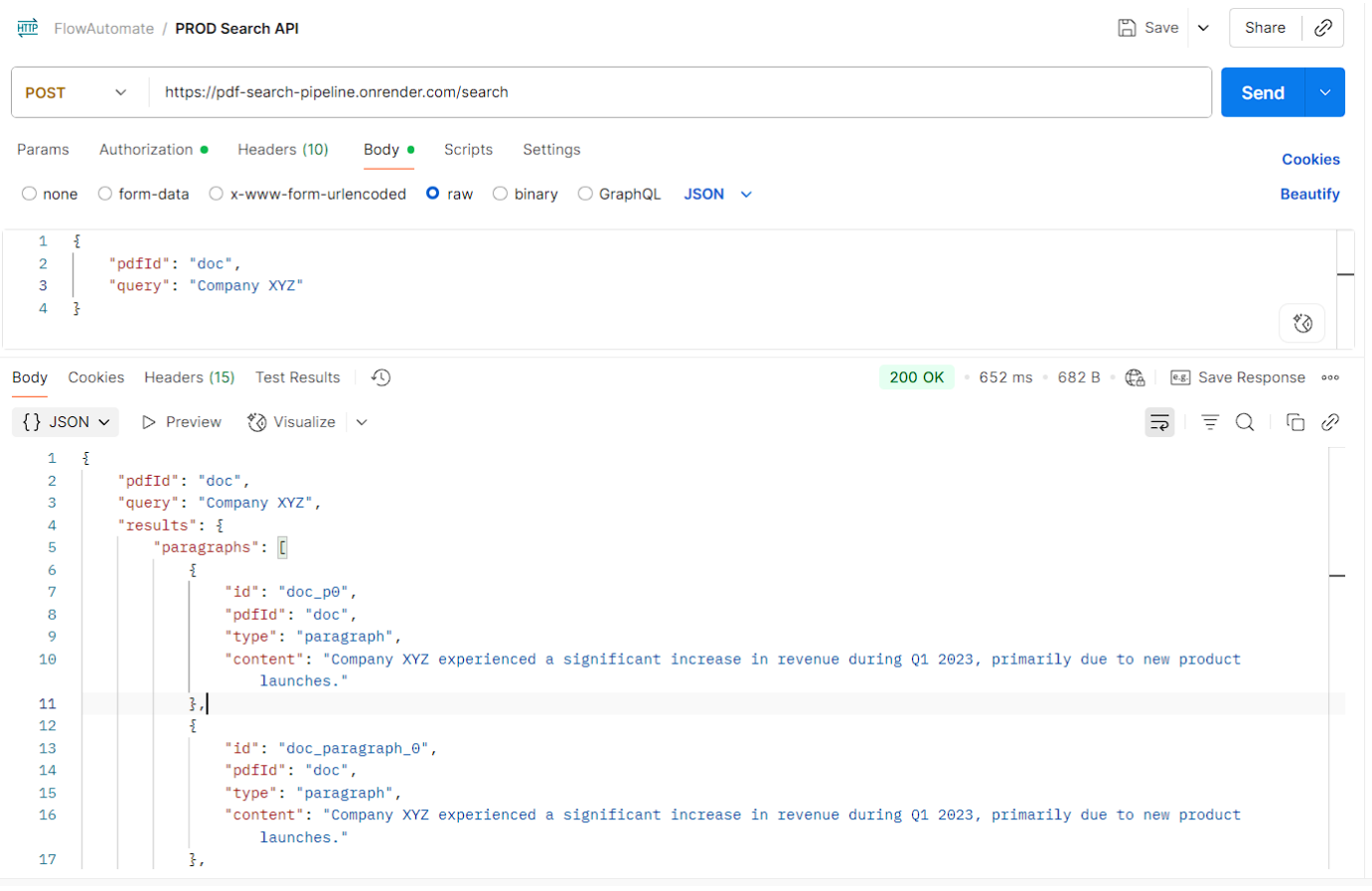

```
{
    "pdfId": "doc",
```

```
    "query": "Company XYZ"  The search keyword or phrase (e.g., sample document).
}
```

- **Response** (Success):

```
{
    "pdfId": "doc",
    "query": "Company XYZ",
    "results": {
        "paragraphs": [
            {
                "id": "doc_p0",
                "pdfId": "doc",
                "type": "paragraph",
                "content": "Company XYZ experienced a significant increase in revenue
during Q1 2023, primarily due to new product launches."
            },
            {
                "id": "doc_p4",
                "pdfId": "doc",
                "type": "paragraph",
                "content": "The company also invested heavily in sustainability
initiatives."
            }
        ],
        "tables": [],
        "images": []
    }
}
```

- **Screenshot**:



🚀 Testing the Deployed APIs

The API is deployed on Render and can be tested directly using the deployed URL:
👉 https://pdf-search-pipeline.onrender.com/

Steps to Test:

1. **Test Ingestion**:
    - Use a tool like Postman or cURL to send a POST /ingest request with the parsed PDF data, including the JWT token in the Authorization header.
    - https://pdf-search-pipeline.onrender.com/ingest
2. **Test Search**:
    - Send a POST /search request with the JWT token and searchable keyword to retrieve matching documents.
    - https://pdf-search-pipeline.onrender.com/search

🔍 Configuring Meilisearch

You can use the deployed Meilisearch instance or set up a local instance using Docker Desktop.

Option 1: Use Deployed Meilisearch

The project is configured to use a deployed Meilisearch instance by default. Update your .env file with the following values from the .env.example:

MEILI_URL=https://meili--meilisearch--6y7qslcbcmz5.code.run

MEILI_MASTER_KEY=F121523A

No additional setup is required to use the deployed Meilisearch instance. Ensure your API requests are sent to the deployed API URL.

Option 2: Set Up Meilisearch Locally with Docker Desktop

If you prefer to run Meilisearch locally, follow these steps:

1. **Install Docker Desktop**:
    - Download and install Docker Desktop following the official instructions: https://www.docker.com/products/docker-desktop/
2. **Run Meilisearch in Docker**:

    docker run -it --rm -p 7700:7700 -e MEILI_MASTER_KEY="your_master_key" getmeili/meilisearch
    - This starts Meilisearch on http://localhost:7700 with data persistence in the ./meili_data directory on your host machine.
    - Replace 'your_master_key' with a secure master key (at least 16 bytes long).
3. **Update Environment Variables**:
    - Modify your .env file to point to the local Meilisearch instance:
    - MEILI_URL=http://localhost:7700

      MEILI_MASTER_KEY=your_master_key

4. **Verify Meilisearch**:
    - Access the Meilisearch instance at http://localhost:7700/health to confirm it's running (should return {"status": "available"}).
5. **Run the API Locally**:
    - Start your Node.js application with npm run dev and test the endpoints as described above.

Notes:

- Mounting volumes (e.g., for data persistence) may cause performance loss and is recommended for development only.
- Meilisearch runs on port 7700 locally.

🛠️ Example .env Configuration

Below is the .env.example file for reference, supporting both deployed and local Meilisearch setups:

```
# MongoDB Configuration

# Use this for MongoDB Atlas (cloud-hosted)

MONGO_URI=mongodb+srv://<your-db-url>

# OR use this for local MongoDB

# MONGO_URI=mongodb://localhost:27017/pdf-search-pipeline


# Meilisearch Configuration

# Use this for the deployed Meilisearch instance

MEILI_URL=https://meili--meilisearch--6y7qslcbcmz5.code.run

MEILI_MASTER_KEY=F121523A

# OR use this for local Meilisearch (Docker)

# MEILI_URL=http://localhost:7700

# MEILI_MASTER_KEY=your_master_key


# API Configuration

API_SECRET_KEY=F121523A

PORT=5000
```