# Elasticsearch Data Extraction Tool – Documentation

## Contents

1. System Requirements ..... 1
2. Setup Instructions ..... 2
3. app.py - Streamlit Web Application ..... 2
4. mine.py - Command-line Data Extraction Script ..... 3
5. Summary ..... 4

This tool consists of two Python scripts to extract and display data from **Elasticsearch**:

- **app.py** → **A Streamlit-based web UI** for interactive data fetching and downloading.
- **mine.py** → **A command-line script** to extract and save data as CSV and JSON.

## 1. System Requirements

**Install Python**

Ensure you have **Python 3.7 or later** installed.

- Check if Python is installed:

*python --version*

- If not installed, download and install from Python's official website.

# 2. Setup Instructions

**Step 1: Install Dependencies**

Run the following command in your terminal or command prompt:

*pip install streamlit elasticsearch pandas python-dotenv*

This will install:

- streamlit → For the UI application
- elasticsearch → To connect with Elasticsearch
- pandas → For data manipulation
- python-dotenv → To securely store credentials

**Step 2: Configure Environment Variables**

Create a **.env** file in the same folder as the scripts and add your Elasticsearch credentials:

ES_HOST= “http://your-elasticsearch-host:9200”

ES_USER= “your_username”

ES_PASSWORD= ”your_password”

Make sure the values match your **Elasticsearch setup**.

# 3. app.py- Streamlit Web Application

**Purpose:**

- Provides an easy-to-use web interface for fetching and downloading data from Elasticsearch.
- Lists all available indices dynamically.
- Allows users to **view** and **download** data in **CSV format**.

**How It Works:**

1. **Loads credentials from .env file**.
2. **Connects to Elasticsearch** with authentication.

3. **Fetches available indices** and displays them in a dropdown menu.
4. **Queries the selected index** and retrieves **up to 1,000 records**.
5. **Displays the data** in a table format.
6. **Allows CSV file download**.

**How to Run:**

Run the following command

*streamlit run app.py*

- This will **open a web application** in your browser where you can: Select an index
  View data
  Download CSV

# 4. mine.py- Command-line Data Extraction Script

**Purpose:**

- Fetches data **without a UI** for batch processing.
- Saves extracted data as **CSV and JSON** files.

**How It Works:**

1. **Loads credentials from .env file**.
2. **Connects to Elasticsearch**.
3. **Checks if the index exists** before querying.
4. **Fetches up to 1,000 records** using a match_all query.
5. **Prints the retrieved data** to the console.
6. **Saves data as:**
   - elasticsearch_data.csv → CSV format
   - elasticsearch_data.json → JSON format

**How to Run:**

Run the following command in your terminal:

*python mine.py*

Once executed, it extracts data and saves it in the **current directory**.

# 5. Summary

| Script | Purpose | Usage |
|---|---|---|
| app.py | Web-based tool for Elasticsearch data extraction | streamlit run app.py |
| mine.py | CLI-based tool to extract and save data | python mine.py |

**Both scripts use .env for secure authentication.**

- **Error handling ensures smooth execution.**
- **Data can be saved in both CSV and JSON formats.**

This documentation provides everything needed to set up, run, and troubleshoot the **Elasticsearch Data Extraction Tool**.

Loom Video : https://www.loom.com/share/80dc880efc90409386058c525677f471?sid=c0010030-7592-4a11-943b-f6f653af128a