API USED

```
import requests

url = "https://jsearch.p.rapidapi.com/search"

querystring = {"query":"cybersecurity","page":"5","num_pages":"1","date_posted":"all"}

headers = {
	"x-rapidapi-key": "09a4dda10fmshab2b1b95642a0e3p1ea115jsn662ff359d59c",
	"x-rapidapi-host": "jsearch.p.rapidapi.com"
}

response = requests.get(url, headers=headers, params=querystring)

print(response.json())
```

Link:
https://rapidapi.com/letscrape-6bRBa3QguO5/api/jsearch/playground/apiendpoint_23823c0b-cab1-4b42-8b13-1ccae3496c07

Cybersecurity Job Scraper

Purpose

The script is designed to automate the process of scraping entry-level cyber security job postings from a web API. It fetches job listings, processes the data, and saves it in two formats: JSON and CSV. This allows users to analyze job data more efficiently.

How It Works

1. **API Request**:
    - The script makes a GET request to the JSearch API using the `requests` library.
    - It sends a query to search for jobs related to "cybersecurity".
    - The API request includes headers for authentication and query parameters to specify the search criteria (e.g., the search term, page number, and number of pages).
2. **Handling the API Response**:
    - After receiving the response, the script checks if the request was successful (HTTP status code 200).
    - It then parses the JSON response and saves it to a file named `jobs_data.json`. This file contains the raw data from the API.
3. **Data Extraction and Cleaning**:
    - The script extracts relevant job details from the JSON data.

    - It performs data cleaning to handle missing or incomplete information and formats the data for easier analysis. This includes:
        - Stripping leading and trailing whitespace from text fields.
        - Handling `None` values and setting default values where necessary.
        - Combining multiple fields (e.g., city, state, and country) into a single location field.
        - Converting lists of skills or categories into comma-separated strings.
4. **Data Preparation for CSV**:
    - It organizes the cleaned job data into a list of dictionaries, each representing a job posting.
    - The data is sorted by job title and company name to ensure a consistent order.
5. **Writing to CSV**:
    - The script writes the formatted job data to a CSV file named `jobs_data.csv`.
    - It uses the `csv` library to create the CSV file with appropriate headers and rows for each job posting.

Key Features

- **Automated Data Retrieval**: Fetches job postings automatically from the API based on predefined search criteria.
- **Data Formatting**: Cleans and formats the raw job data to make it suitable for analysis.
- **Multi-format Output**: Provides output in both JSON and CSV formats for flexibility in data usage and analysis.

Use Cases

- **Job Market Analysis**: Users can analyze trends in entry-level cyber security job postings, such as the most common job titles, required skills, and salary ranges.
- **Resume Tailoring**: Job seekers can use the data to tailor their resumes and cover letters according to the most frequently required qualifications and skills.
- **Recruitment**: Recruiters can use the data to understand the job market and adjust their recruitment strategies.

This script provides a practical tool for anyone interested in cybersecurity job market data, offering insights through automated data scraping and processing.