

# ListJob Project

Monday, January 18, 2021 11:39 PM

## Problem Definition:

Write a program to extract the skill sets and corresponding jobs with that skill set in listed  
The skill set you are looking for is "engineering and technology"

Output: list of all the jobs that have this skill in the job requirement

For this exercise use the public api : <http://api.dataatwork.org/v1/>

The api is explained here <https://github.com/workforce-data-initiative/skills-api/wiki/API-Overview>

Make use of the skills api <https://github.com/workforce-data-initiative/skills-api/wiki/API-Overview#4-skill-name-autocomplete>

Then make use of the related job skills api <https://github.com/workforce-data-initiative/skills-api/wiki/API-Overview#9-get-associated-jobs-for-a-skill>

Explore the api and try it out here <http://api.dataatwork.org/v1/spec/>

\*\*\*\*Bonus: Create an array of all the jobs with the skill and push the payload to a requestbin (<http://requestbin.net/>).

Git URL: [https://github.com/KrithiSri-Git/Python\\_Projects](https://github.com/KrithiSri-Git/Python_Projects)

Script Name : *getjoblistenggtech.py*

Script extracts the job list using api into an array based on the skill 'engineering and technology' and posts the result to requestbin.

## Approach taken to solve the problem

1. Analyze the data set and API capabilities
2. GET Skills with the skill name : 'engineering and technology' using the GET API Call
3. Use data dictionary `dict_skillsfilter` to store the results and print the output
4. For Each skill record having 'engineering and technology'
  - a. From the JSON response identify the UUID for the 'engineering and technology' skill set
  - b. Use the UUID to get the related jobs using the GET API call
  - c. Use data dictionary `dict_joblist` to store the results and print the output
  - d. Parse through the data and collect the 'job titles' into an array `array_joblist`
5. Print the job list in the array (`array_joblist`)
6. Construct the POST message with the array data and post the result to requestbin

## Output Snapshot

```
pythonProject - getjoblistenggtech.py
21 resp_joblist = requests.get("http://api.dataatwork.org/v1/skills/" + skill_uuid_engg_tech + "/related_jobs")
22 data_joblist = resp_joblist.text
23 dict_joblist=json.loads(data_joblist)
24 #print(json.dumps(dict_joblist, indent=4))
25 print("Number of Jobs posted :", len(dict_joblist['jobs']))
26
27
28 print("-----")
29 print(" List of Job Titles ")
30 print("-----")
31
32 #Store the job list in the array
33 for i in dict_joblist['jobs']:
34     array_joblist.append(i['job_title'])
35
36 #End of For Loop
37
38 # Print the array Joblist with the number of records
39 print(len(array_joblist))
40 print(array_joblist)
41 headers={"Title": "Engineering Jobs"}
42 data=json.dumps(array_joblist)
```

Run: getjoblistenggtech

```
C:\Users\KRITHIGA S\PycharmProjects\pythonProject\venv\scripts\python.exe "C:\Users\KRITHIGA S\PycharmProjects\pythonProject\getjoblistenggtech.py"
[
  {
    "uuid": "c09c8f8c163907ac25a37c3dd591ba2e",
    "suggestion": "engineering and technology",
    "normalized_skill_name": "engineering and technology"
  }
]
UUID for 'engineering and technology': c09c8f8c163907ac25a37c3dd591ba2e
Number of Jobs posted : 954
-----
List of Job Titles
-----
954
['Computer Hardware Engineers', 'Electronics Engineers, Except Computer', 'Automotive Engineers', 'Aerospace Engineering and Operations Technicians', 'Mechanical Engineers', 'Chemical Engineers', 'Architectural and Engineering'
{"success":true}]

Process finished with exit code 0
```

## Request Bin

<https://en4832wfm1s7a.x.pipedream.net/>

The screenshot shows the Request Bin web interface. The browser address bar displays the endpoint URL: `requestbin.com/en4832wfm1s7a/1nHAsbZtH2iZepwICg980LEa`. The interface includes a sidebar with a search bar and a main content area. The main content area displays the endpoint URL: `https://en4832wfm1s7a.x.pipedream.net/`. Below the URL, there is a section titled "Generate Test Events" with a button labeled "Generate Test Events". A message states: "Send requests to this endpoint to inspect webhooks, custom events and more:". Below this, there are three bullet points: "Events appear instantly", "Endpoints don't expire", and "Easily delete a single event, or all". A code block shows a curl command: `curl -d '{"name": "Darth Vader"}' -H 'Content-Type: application/json' https://en4832wfm1s7a.x.pipedream.net/`. At the bottom, there is a section titled "NEW" with a heading "Preview the new Request Bin — now with an API and more sources!". Below this, there are two buttons: "Create HTTP Source" and "View on Github". A list of features is provided: "Create bins and inspect events from your browser or via APIs", "Generate events from HTTP requests, RSS feeds, Tweets, Airtable records, Github events, and more!", and "Trigger serverless Node.js workflows on every event with integrated auth for 100+ apps".

## Output posted to requestbin

The screenshot shows the Request Bin web interface displaying the output of a POST request. The browser address bar displays the endpoint URL: `requestbin.com/en4832wfm1s7a/1nH8ynAFRUFxAqd7EYsGg338r`. The interface includes a sidebar with a search bar and a main content area. The main content area displays the output of a POST request. The request details are shown, including the method (POST) and the URL (/). The headers section lists the following headers: `host: en4832wfm1s7a.x.pipedream.net`, `accept: */*`, `accept-encoding: gzip, deflate`, `title: Engineering Jobs`, `user-agent: python-requests/2.25.1`, `content-length: 34341`, and `connection: keep-alive`. The body section shows the JSON response, which is a list of engineering jobs. The response is displayed in a structured format, showing the root of the JSON object and the list of jobs. The jobs are listed as follows: `0: "Computer Hardware Engineers"`, `1: "Electronics Engineers, Except Computer"`, `2: "Automotive Engineers"`, `3: "Aerospace Engineering and Operations Technicians"`, `4: "Mechanical Engineers"`, `5: "Chemical Engineers"`, `6: "Architectural and Engineering Managers"`, `7: "Nuclear Engineers"`, `8: "Transportation Engineers"`, `9: "Wind Energy Engineers"`, `10: "Mechatronics Engineers"`, `11: "Electrical Engineering Technologists"`, `12: "Agricultural Engineers"`, `13: "Petroleum Engineers"`, `14: "Aerospace Engineers"`, `15: "Water/Wastewater Engineers"`, `16: "Manufacturing Engineering Technologists"`, `17: "Nanosystems Engineers"`, `18: "Fuel Cell Engineers"`, `19: "Photonics Engineers"`, `20: "Manufacturing Engineers"`, `21: "Mechanical Engineering Technologists"`, and `22: "Electrical Engineers"`. The interface also includes a sidebar with a search bar and a main content area. The sidebar shows the date and time of the request: "Today 19:23:12 PM POST /". The main content area shows the request details and the response body. The response body is displayed in a structured format, showing the root of the JSON object and the list of jobs. The jobs are listed as follows: `0: "Computer Hardware Engineers"`, `1: "Electronics Engineers, Except Computer"`, `2: "Automotive Engineers"`, `3: "Aerospace Engineering and Operations Technicians"`, `4: "Mechanical Engineers"`, `5: "Chemical Engineers"`, `6: "Architectural and Engineering Managers"`, `7: "Nuclear Engineers"`, `8: "Transportation Engineers"`, `9: "Wind Energy Engineers"`, `10: "Mechatronics Engineers"`, `11: "Electrical Engineering Technologists"`, `12: "Agricultural Engineers"`, `13: "Petroleum Engineers"`, `14: "Aerospace Engineers"`, `15: "Water/Wastewater Engineers"`, `16: "Manufacturing Engineering Technologists"`, `17: "Nanosystems Engineers"`, `18: "Fuel Cell Engineers"`, `19: "Photonics Engineers"`, `20: "Manufacturing Engineers"`, `21: "Mechanical Engineering Technologists"`, and `22: "Electrical Engineers"`.

RequestBin.com — A modern re... x +

requestbin.com/r/en4832wfm1s7a/1nHBynAFRUFxAqud7EYyGg338r

Apps Templates DL with TubeOffline tutorials oomwatch Sharepoint Introduction to Mar... Chrome Web Store... Gmail Vmware free up space CUSD Distance Lear...

pipedream Sign in to old version

▼Untitled publicEndpointhttps://en4832wfm1s7a.x.pipedream.net/ Copy + New

LIVEPAUSEType to search...

Today19:23 12 PMPOST /

929: "Payroll and Timekeeping Clerks"

930: "Credit Checkers"

931: "Order Clerks"

932: "Credit Counselors"

933: "Models"

934: "Municipal Clerks"

935: "Eligibility Interviewers, Government Programs"

936: "Billing, Cost, and Rate Clerks"

937: "Judicial Law Clerks"

938: "Travel Guides"

939: "Education Administrators, Preschool and Childcare Center/Program"

940: "Fundraisers"

941: "Library Technicians"

942: "Crossing Guards"

943: "Door-To-Door Sales Workers, News and Street Vendors, and Related Workers"

944: "Correspondence Clerks"

945: "Proofreaders and Copy Markers"

946: "Court Clerks"

947: "Bus Drivers, School or Special Client"

948: "Postal Service Mail Sorters, Processors, and Processing Machine Operators"

949: "Law Teachers, Postsecondary"

950: "History Teachers, Postsecondary"

951: "Loan Counselors"

952: "Marriage and Family Therapists"

953: "Child, Family, and School Social Workers"

NEW

Preview the new Request Bin — now with an API and more sources!

Create HTTP SourceView on Github

- Create bins and inspect events from your browser or via APIs
- Generate events from HTTP requests , RSS feeds , Tweets , Airtable records , Github events , and more!
- Trigger serverless Node.js workflows on every event with integrated auth for 100+ apps

DELETE ALL