

# Crustdata Discovery And Enrichment API

[Introduction](#)[Getting Started](#)[Obtaining Authorization Token](#)[Data Dictionary](#)[Company Endpoints](#)[Enrichment: Company Data API](#)[Company Discovery: Screening API](#)[Additional examples](#)[Company Identification API](#)[Company Dataset API](#)[All dataset endpoints](#)[Search: LinkedIn Company Search API \(real-time\)](#)[Building the Company/People Search Criteria Filter](#)[Making Requests](#)[LinkedIn Posts by Company API \(real-time\)](#)[LinkedIn Posts Keyword Search \(real-time\)](#)[People Endpoints](#)[Enrichment: People Profile\(s\) API](#)[Search: LinkedIn People Search API \(real-time\)](#)[Making Requests](#)[LinkedIn Posts by Person API \(real-time\)](#)[LinkedIn Posts Keyword Search \(real-time\)](#)[API Usage Endpoints](#)[Get remaining credits](#)

# Introduction

The Crustdata API gives you programmatic access to firmographic and growth metrics data for companies across the world from more than 16 datasets (Linkedin headcount, Glassdoor, Instagram, G2, Web Traffic, Apple App Store reviews, Google Play Store, News among others).

This documentation describes various available API calls and schema of the response. If you have any questions, please reach out to [abhilash@crustdata.com](mailto:abhilash@crustdata.com).

# Getting Started

## Obtaining Authorization Token

Reach out to [abhilash@crustdata.com](mailto:abhilash@crustdata.com) get an authorization token (API key) .

# Data Dictionary

 [Crustdata Data Dictionary](#)

# Company Endpoints

## Enrichment: Company Data API

**Overview:** This endpoint enriches company data by retrieving detailed information about one or multiple companies using either their domain, name, or ID.

Required: authentication token `auth_token` for authorization.

## ▼ Request

### Parameters

- **company\_domain:** *string* (comma-separated list, up to 25 domains)
  - **Description:** The domain(s) of the company(ies) you want to retrieve data for.
  - **Example:** `company_domain=hubspot.com,google.com`
- **company\_name:** *string* (comma-separated list, up to 25 names; use double quotes if names contain commas)
  - **Description:** The name(s) of the company(ies) you want to retrieve data for.
  - **Example:** `company_name="Acme, Inc.", "Widget Co"`
- **company\_linkedin\_url:** *string* (comma-separated list, up to 25 URLs)
  - **Description:** The LinkedIn URL(s) of the company(ies).
  - **Example:**  
`company_linkedin_url=https://linkedin.com/company/hubspot,https://link  
edin.com/company/clay-hq`
- **company\_id:** *integer* (comma-separated list, up to 25 IDs)
  - **Description:** The unique ID(s) of the company(ies) you want to retrieve data for.
  - **Example:** `company_id=12345,67890`
- **fields:** *string* (comma-separated list of fields)
  - **Description:** Specifies the fields you want to include in the response.  
Supports nested fields up to a certain level.
  - **Example:**  
`fields=company_name,company_domain,glassdoor.glassdoor_review_count`
- **enrich\_realtime:** *boolean* (False by default)
  - Description: When True and the requested company is not present in Crustdata's database, the company is enriched within 10 minutes of the request

## Using the `fields` Parameter

The `fields` parameter allows you to customize the response by specifying exactly which fields you want to retrieve. This can help reduce payload size and improve performance.

## Important Notes

- **Nested Fields:** You can specify nested fields up to the levels defined in the response structure (see [Field Structure](#) below). Fields nested beyond the allowed levels or within lists (arrays) cannot be individually accessed.
- **Default Fields:**
  - **Top-Level Non-Object Fields:** If you do not specify the `fields` parameter, the response will include all top-level non-object fields by default (e.g., `company_name`, `company_id`).
  - **Object Fields:** By default, the response **will not include** object fields like `decision_makers` and `founders.profiles`, even if you have access to them. To include these fields, you must explicitly specify them using the `fields` parameter.
- **User Permissions:** Access to certain fields may be restricted based on your user permissions. If you request fields you do not have access to, the API will return an error indicating unauthorized access.

## Examples

### ▼ Request by Company Domain:

- **Use Case:** Ideal for users who have one or more company website domains and need to fetch detailed profiles.
- **Note:** You can provide up to 25 domains in a comma-separated list.
- **Request:**

```
curl 'https://api.crustdata.com/screener/company?company_domain=hubspot.com,google.com' \ --header 'Accept: application/json, text/plain, */*' \ --header 'Accept-Language: en-US,en;q=0.9' \ --header 'Authorization: Token $token'
```

## ▼ Request by Company Name:

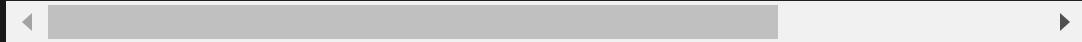
- **Use Case:** Suitable for users who have one or more company names and need to retrieve detailed profiles.
- **Note:** You can provide up to 25 names in a comma-separated list. If a company name contains a comma, enclose the name in double quotes.
- **Request:**

```
curl 'https://api.crustdata.com/screener/company?company_name="Hu  
bSpot","Google, Inc."' \ --header 'Accept: application/json, tex  
t/plain, */*' \ --header 'Accept-Language: en-US,en;q=0.9' \ --he  
ader 'Authorization: Token $token'
```

## ▼ Request by Company LinkedIn URL:

- **Use Case:** Suitable for users who have one or more company LinkedIn urls and need to retrieve detailed profiles.
- **Note:** You can provide up to 25 names in a comma-separated list. If a company name contains a comma, enclose the name in double quotes.
- **Request:**

```
curl 'https://api.crustdata.com/screener/company?  
company_linkedin_url=https://linkedin.com/company/hubspot,https://li  
nkedin.com/company/hubspot,hubspot' \ --header 'Accept: application/json, text/plain, */*' \ --heade  
r 'Accept-Language: en-US,en;q=0.9' \ --header 'Authorization: Token $token'
```



## ▼ Request by Company ID:

- **Use Case:** Suitable for users who have ingested one or more companies from Crustdata already and want to enrich their data by Crustdata's `company_id`. Users generally use this when they want time-series data for specific companies after obtaining the `company_id` from the `screening` endpoint.
- **Note:** You can provide up to 25 IDs in a comma-separated list.
- **Request:**

```
curl 'https://api.crustdata.com/screener/company?  
company_id=631480,789001' \ --header 'Accept: application/json,  
text/plain, */*' \ --header 'Accept-Language: en-US,en;q=0.9' \ -  
-header 'Authorization: Token $token'
```

## ▼ Request with Specific Fields

- **Use Case:** Fetch only specific fields to tailor the response to your needs.
- **Request**

```
curl 'https://api.crustdata.com/screener/company?  
company_domain=swiggy.com&fields=company_name,headcount.linkedin_he  
\ --header 'Authorization: Token $token' \ --header 'Accept:  
application/json'
```

- More examples of Using `fields` parameter

## ▼ Request with Realtime Enrichment

- **Use Case:** For companies not tracked by Crustdata, you want to enrich them within 10 minutes of the request

```
curl --location 'https://api.crustdata.com/screener/company?  
company_linkedin_url=https://www.linkedin.com/company/usebramble&enrich_  
\ --header 'Accept: application/json, text/plain, /' \ --header 'Accept-  
US,en;q=0.9' \ --header 'Authorization: Token $token'
```



## ▼ Response Structure

The response is a JSON array containing company objects. Below is the structure of the response up to the levels where you can filter using the `fields` parameter.

## Top-Level Fields

- `company_id`: *integer*
- `company_name`: *string*
- `linkedin_profile_url`: *string*
- `linkedin_id`: *string*
- `linkedin_logo_url`: *string*
- `company_twitter_url`: *string*
- `company_website_domain`: *string*
- `hq_country`: *string*
- `headquarters`: *string*
- `largest_headcount_country`: *string*
- `hq_street_address`: *string*
- `company_website`: *string*
- `year_founded`: *string* (ISO 8601 date)
- `fiscal_year_end`: *string*
- `estimated_revenue_lower_bound_usd`: *integer*
- `estimated_revenue_higher_bound_usd`: *integer*
- `employee_count_range`: *string*
- `company_type`: *string*
- `linkedin_company_description`: *string*
- `acquisition_status`: *string or null*
- `ceo_location`: *string*

## Nested Objects

You can filter up to the following nested levels:

### **all\_office\_addresses**

- *array of strings*

### **markets**

- *array of strings*

### **stock\_symbols**

- *array of strings*

### **taxonomy**

- **linkedin\_specialties**: *array of strings*
- **linkedin\_industries**: *array of strings*
- **crunchbase\_categories**: *array of strings*

### **competitors**

- **competitor\_website\_domains**: *array of strings or null*
- **paid\_seo\_competitors\_website\_domains**: *array of strings*
- **organic\_seo\_competitors\_website\_domains**: *array of strings*

### **headcount**

- **linkedin\_headcount**: *integer*
- **linkedin\_headcount\_total\_growth\_percent**
  - **mom**: *float*
  - **qoq**: *float*
  - **six\_months**: *float*
  - **yoy**: *float*
  - **two\_years**: *float*

- `linkedin_headcount_total_growth_absolute`
  - `mom`: *float*
  - `qoq`: *float*
  - `six_months`: *float*
  - `yoy`: *float*
  - `two_years`: *float*
- `linkedin_headcount_by_role_absolute`: *object*
- `linkedin_headcount_by_role_percent`: *object*
- `linkedin_role_metrics`
  - `all_roles`: *string*
  - `0_to_10_percent`: *string*
  - `11_to_30_percent`: *string*
  - `31_to_50_percent`: *string or null*
  - `51_to_70_percent`: *string or null*
  - `71_to_100_percent`: *string or null*
- `linkedin_headcount_by_role_six_months_growth_percent`: *object*
- `linkedin_headcount_by_role_yoy_growth_percent`: *object*
- `linkedin_headcount_by_region_absolute`: *object*
- `linkedin_headcount_by_region_percent`: *object*
- `linkedin_region_metrics`
  - `all_regions`: *string*
  - `0_to_10_percent`: *string*
  - `11_to_30_percent`: *string*
  - `31_to_50_percent`: *string or null*
  - `51_to_70_percent`: *string or null*
  - `71_to_100_percent`: *string or null*
- `linkedin_headcount_by_skill_absolute`: *object*
- `linkedin_headcount_by_skill_percent`: *object*

- **linkedin\_skill\_metrics**
  - **all\_skills**: *string*
  - **0\_to\_10\_percent**: *string or null*
  - **11\_to\_30\_percent**: *string*
  - **31\_to\_50\_percent**: *string or null*
  - **51\_to\_70\_percent**: *string or null*
  - **71\_to\_100\_percent**: *string or null*
- **linkedin\_headcount\_timeseries**: *array of objects* (Cannot filter within this array)
- **linkedin\_headcount\_by\_function\_timeseries**: *object* (Cannot filter within this object)

## web\_traffic

- **monthly\_visitors**: *integer*
- **monthly\_visitor\_mom\_pct**: *float*
- **monthly\_visitor\_qoq\_pct**: *float*
- **traffic\_source\_social\_pct**: *float*
- **traffic\_source\_search\_pct**: *float*
- **traffic\_source\_direct\_pct**: *float*
- **traffic\_source\_paid\_referral\_pct**: *float*
- **traffic\_source\_referral\_pct**: *float*
- **monthly\_visitors\_timeseries**: *array of objects* (Cannot filter within this array)
- **traffic\_source\_social\_pct\_timeseries**: *array of objects* (Cannot filter within this array)
- **traffic\_source\_search\_pct\_timeseries**: *array of objects* (Cannot filter within this array)
- **traffic\_source\_direct\_pct\_timeseries**: *array of objects* (Cannot filter within this array)
- **traffic\_source\_paid\_referral\_pct\_timeseries**: *array of objects* (Cannot filter within this array)

- **traffic\_source\_referral\_pct\_timeseries**: *array of objects* (Cannot filter within this array)

## glassdoor

- **glassdoor\_overall\_rating**: *float*
- **glassdoor\_ceo\_approval\_pct**: *integer*
- **glassdoor\_business\_outlook\_pct**: *integer*
- **glassdoor\_review\_count**: *integer*
- **glassdoor\_senior\_management\_rating**: *float*
- **glassdoor\_compensation\_rating**: *float*
- **glassdoor\_career\_opportunities\_rating**: *float*
- **glassdoor\_culture\_rating**: *float or null*
- **glassdoor\_diversity\_rating**: *float or null*
- **glassdoor\_work\_life\_balance\_rating**: *float or null*
- **glassdoor\_recommend\_to\_friend\_pct**: *integer or null*
- **glassdoor\_ceo\_approval\_growth\_percent**
  - **mom**: *float*
  - **qoq**: *float*
  - **yoy**: *float*
- **glassdoor\_review\_count\_growth\_percent**
  - **mom**: *float*
  - **qoq**: *float*
  - **yoy**: *float*

## g2

- **g2\_review\_count**: *integer*
- **g2\_average\_rating**: *float*
- **g2\_review\_count\_mom\_pct**: *float*
- **g2\_review\_count\_qoq\_pct**: *float*

- `g2_review_count_yoy_pct`: *float*

## linkedin\_followers

- `linkedin_followers`: *integer*
- `linkedin_follower_count_timeseries`: *array of objects* (Cannot filter within this array)
- `linkedin_followers_mom_percent`: *float*
- `linkedin_followers_qoq_percent`: *float*
- `linkedin_followers_six_months_growth_percent`: *float*
- `linkedin_followers_yoy_percent`: *float*

## funding\_and\_investment

- `crunchbase_total_investment_usd`: *integer*
- `days_since_last_fundraise`: *integer*
- `last_funding_round_type`: *string*
- `crunchbase_investors`: *array of strings*
- `last_funding_round_investment_usd`: *integer*
- `funding_milestones_timeseries`: *array of objects* (Cannot filter within this array)

## job\_openings

- `recent_job_openings_title`: *string or null*
- `job_openings_count`: *integer or null*
- `job_openings_count_growth_percent`
  - `mom`: *float or null*
  - `qoq`: *float or null*
  - `yoy`: *float or null*
- `job_openings_by_function_qoq_pct`: *object*
- `job_openings_by_function_six_months_growth_pct`: *object*
- `open_jobs_timeseries`: *array of objects* (Cannot filter within this array)
- `recent_job_openings`: *array of objects* (Cannot filter within this array)

## seo

- `average_seo_organic_rank`: *integer*
- `monthly_paid_clicks`: *integer*
- `monthly_organic_clicks`: *integer*
- `average_ad_rank`: *integer*
- `total_organic_results`: *integer or float*
- `monthly_google_ads_budget`: *integer or float*
- `monthly_organic_value`: *integer*
- `total_ads_purchased`: *integer*
- `lost_ranked_seo_keywords`: *integer*
- `gained_ranked_seo_keywords`: *integer*
- `newly_ranked_seo_keywords`: *integer*

## founders

- `founders_locations`: *array of strings*
- `founders_education_institute`: *array of strings*
- `founders_degree_name`: *array of strings*
- `founders_previous_companies`: *array of strings*
- `founders_previous_titles`: *array of strings*
- `profiles`: *array of objects* (Cannot filter within this array)

## decision\_makers

- `decision_makers`: *array of objects* (Cannot filter within this array)

## news\_articles

- `news_articles`: *array of objects* (Cannot filter within this array)

## ▼ Response

### Examples

The response provides a comprehensive profile of the company, including firmographic details, social media links, headcount data, and growth metrics.

For a detailed response data structure, refer to this JSON  [json\\_hero](#) [JSON Hero](#)

## ▼ Key Points

### Credits

- Database Enrichment:
  - 1 credits per company.
- Real-Time Enrichment (`enrich_realtime=True`):
  - 4+1 credits per company.

### Enrichment Status

When you request data for a company not in our database, we start an enrichment process that takes up to **24 hours** (or **10 minutes** if `enrich_realtime` is `true` ).

The API response includes a `status` field:

- `enriching` : The company is being processed, poll later to get the full company info
- `not_found` : Enrichment failed (e.g., no website or employees). You can stop polling for this company.

```
[ { "status": "enriching", "message": "The following companies will be enriched in the next 24 hours", "companies": [ { "identifier": "https://www.linkedin.com/company/123456", "type": "linkedin_url" } ] } ]
```

### Limitations on Nested Fields

- Maximum Nesting Level: You can specify nested fields **only up to the levels defined above**

- **Default Exclusion of Certain Fields:** Even if you have access to fields like `decision_makers` and `founders.profiles`, they will not be included in the response by default when the `fields` parameter is not provided. You must explicitly request these fields using the `fields` parameter.

- **Example:**

```
# Will not include 'decision_makers' or 'founders.profiles' by default
curl 'https://api.crustdata.com/screener/company?company_id=123' \
--header 'Authorization: Token $token' \
--header 'Accept: application/json'
```

To include them, specify in `fields`:

```
curl 'https://api.crustdata.com/screener/company?company_id=123&fields=decision_makers,founders.profiles' \
--header 'Authorization: Token $token' \
--header 'Accept: application/json'
```

- **Unavailable Fields:** If you request a field that is not available or beyond the allowed nesting level, the API will return an error indicating that the field is not available for filtering.

## Company Discovery: Screening API

**Overview:** The company screening API request allows you to screen and filter companies based on various growth and firmographic criteria.

Required: authentication token `auth_token` for authorization.

## ▼ Request

In the example below, we get companies that meet the following criteria:

- Have raised > \$5,000,000 in total funding AND
- Have headcount > 50 AND
- Have largest headcount country as USA

## ▼ cURL

```
curl 'https://api.crustdata.com/screener/screen/' \ -H 'Accept: application/json, text/plain, /' \ -H 'Accept-Language: en-US,en;q=0.9' \ -H 'Authorization: Token $auth_token' \ -H 'Connection: keep-alive' \ -H 'Content-Type: application/json' \ -H 'Origin: https://crustdata.com' \ --data-raw '{ "metrics": [ { "metric_name": "linkedin_headcount_and_glassdoor_ceo_approval_and_g2" } ], "filters": { "op": "and", "conditions": [ { "column": "crunchbase_total_investment_usd", "type": ">", "value": 5000000, "allow_null": false }, { "column": "linkedin_headcount", "type": ">", "value": 50, "allow_null": false }, { "column": "largest_headcount_country", "type": "(.)", "value": "USA", "allow_null": false } ] }, "hidden_columns": [], "offset": 0, "count": 100, "sorts": [] }' \ --compressed
```

## ▼ Python

```
import requests
headers = {
    'Accept': 'application/json, text/plain, /',
    'Accept-Language': 'en-US,en;q=0.9',
    'Authorization': 'Token $auth_token', # replace $auth_token
    'Connection': 'keep-alive',
    'Content-Type': 'application/json',
    'Origin': 'https://crustdata.com'
}
json_data = {
    'metrics': [
        {
            'metric_name': 'linkedin_headcount_and_glassdoor_ceo_approval_and_g2'
        }
    ],
    'filters': {
        'op': 'and',
        'conditions': [
            {
                'column': 'crunchbase_total_investment_usd',
                'type': '=>',
                'value': 5000000,
                'allow_null': False
            },
            {
                'column': 'linkedin_headcount',
                'type': '=>',
                'value': 50,
                'allow_null': False
            },
            {
                'column': 'largest_headcount_country',
                'type': '(.)',
                'value': 'USA',
                'allow_null': False
            }
        ]
    },
    'hidden_columns': [],
    'offset': 0,
    'count': 100,
    'sorts': []
}
response = requests.post('https://api.crustdata.com/screener/screen/',
                        headers=headers,
                        json=json_data)
```

## ▼ Request Body Overview

The request body is a JSON object that contains the following parameters:

Parameter	Description
metrics	An array of metric objects containing the metric name. Value should always be <code>[{"metric_name": "linkedin_headcount_and_glassdoor_ceo_approval_and_g2"}]</code>
filters	An object containing the filter conditions.
offset	The starting point of the result set. Default value is 0.
count	The number of results to return in a single request. Maximum value is <code>100</code> . Default value is <code>100</code> .
sorts	An array of sorting criteria.

### Parameters:

- `metrics`

Dictates the columns in the response. The only possible value is

```
[{"metric_name": "linkedin_headcount_and_glassdoor_ceo_approval_and_g2"}]
```

- **filters**

Example:

```
{ "op": "and", "conditions": [ { "op": "or", "conditions": [ { "hq_country": "USA", "type": "(.)", "value": "USA"}, { "hq_country": "IND", "type": "(.)", "value": "IND"} ] }, { "column": "crunchbase_total_investment_usd", "type": "=>", "value": "50000000"}, { "column": "largest_headcount_country", "type": "(.)", "value": "USA"} ] }
```

The filters object contains the following parameters:

Parameter	Description	Required
op	The operator to apply on the conditions. The value can be <b>"and"</b> or <b>"or"</b> .	Yes
conditions	An array of complex filter objects or basic filter objects (see below)	Yes

- **conditions** parameter

This has two possible types of values

### 1. Basic Filter Object

Example: `{"column": "linkedin_headcount", "type": ">", "value": "50" }`

The object contains the following parameters:

Parameter	Description	Required
column	The name of the column to filter.	Yes
type	The filter type. The value can be "=>", "=<", "=", "!="; "in", "(.)", "[.]"	Yes
value	The filter value.	Yes
allow_null	Whether to allow null values. The value can be "true" or "false". Default value is "false".	No

- ▶ List of all `column` values
- ▶ List of all `type` values

### 2. Complex Filter Object

Example:

```
{ "op": "or", "conditions": [ { "hq_country": "USA"}, { "hq_country": "IND"} ] }
```

Same schema as the parent `filters` parameter

## ▼ Response

Example:  json\_hero JSON Hero

The response is JSON object that consists of two main components: `fields` and `rows`.

- `fields`: An array of objects representing the columns in the dataset.
- `rows`: An array of arrays, each representing a row of data.

The values in each of the `rows` elements are ordered in the same sequence as the fields in the `fields` array. For example, the `i`th value in a row corresponds to the `i`th field in the `fields` array.

## ▼ Parsing the response

Given the following response object

```
{ "fields": [ {"type": "string", "api_name": "company_name", "hidden": false}, {"type": "number", "api_name": "valuation_usd", "hidden": false}, {"type": "number", "api_name": "crunchbase_total_investment_usd", "hidden": false}, {"type": "string", "api_name": "markets", "hidden": false}, {"type": "number", "api_name": "days_since_last_fundraise", "hidden": false}, {"type": "number", "api_name": "linkedin_headcount", "hidden": false}, {"type": "number", "api_name": "linkedin_headcount_mom_percent", "hidden": false} ], "rows": [ ["Sketch", null, 20000000, "PRIVATE", 1619, 258, -11.64] ] }
```

The first element in `rows` (i.e. `"Sketch"`) corresponds to `fields[0][“api_name”]` (i.e. `"company_name"`).

The second element in `rows` (i.e. `null`) corresponds to `fields[1][“api_name”]` (i.e. `"valuation_usd"`), and so on.

### Pseudo code for mapping `fields` → `rows[i]`

Here's a pseudo code to help understand this mapping:

```
for each row in rows: for i in range(length(row)): field_name = fields[i][“api_name”] field_value = row[i] # Map field_name to field_value
```

In simple terms:

- For each row, iterate over each value.
- Map the `i` th value of the row to the `i` th `api_name` in the fields.

Here is the complete list of fields in the response for each company

▼ Complete list of columns

1. company\_name
2. company\_website
3. company\_website\_domain
4. linkedin\_profile\_url
5. monthly\_visitors
6. valuation\_usd
7. crunchbase\_total\_investment\_usd
8. markets
9. days\_since\_last\_fundraise
10. linkedin\_headcount
11. linkedin\_headcount\_mom\_percent
12. linkedin\_headcount\_qoq\_percent
13. linkedin\_headcount\_yoy\_percent
14. linkedin\_headcount\_mom\_absolute
15. linkedin\_headcount\_qoq\_absolute
16. linkedin\_headcount\_yoy\_absolute
17. glassdoor\_overall\_rating
18. glassdoor\_ceo\_approval\_pct
19. glassdoor\_business\_outlook\_pct
20. glassdoor\_review\_count
21. g2\_review\_count
22. g2\_average\_rating
23. company\_id
24. hq\_country
25. headquarters
26. largest\_headcount\_country
27. last\_funding\_round\_type

28. valuation\_date
29. linkedin\_categories
30. linkedin\_industries
31. crunchbase\_investors
32. crunchbase\_categories
33. acquisition\_status
34. company\_year\_founded
35. technology\_domains
36. founder\_names\_and\_profile\_urls
37. founders\_location
38. ceo\_location
39. founders\_education\_institute
40. founders\_degree\_name
41. founders\_previous\_company
42. founders\_previous\_title
43. monthly\_visitor\_mom\_pct
44. monthly\_visitor\_qoq\_pct
45. traffic\_source\_social\_pct
46. traffic\_source\_search\_pct
47. traffic\_source\_direct\_pct
48. traffic\_source\_paid\_referral\_pct
49. traffic\_source\_referral\_pct
50. meta\_total\_ads
51. meta\_active\_ads
52. meta\_ad\_platforms
53. meta\_ad\_url
54. meta\_ad\_id
55. average\_organic\_rank

- 56. monthly\_paid\_clicks
- 57. monthly\_organic\_clicks
- 58. average\_ad\_rank
- 59. total\_organic\_results
- 60. monthly\_google\_ads\_budget
- 61. monthly\_organic\_value
- 62. total\_ads\_purchased
- 63. lost\_ranks
- 64. gained\_ranks
- 65. newly\_ranked
- 66. paid\_competitors
- 67. organic\_competitors
- 68. linkedin\_followers
- 69. linkedin\_headcount\_engineering
- 70. linkedin\_headcount\_sales
- 71. linkedin\_headcount\_operations
- 72. linkedin\_headcount\_human\_resource
- 73. linkedin\_headcount\_india
- 74. linkedin\_headcount\_usa
- 75. linkedin\_headcount\_engineering\_percent
- 76. linkedin\_headcount\_sales\_percent
- 77. linkedin\_headcount\_operations\_percent
- 78. linkedin\_headcount\_human\_resource\_percent
- 79. linkedin\_headcount\_india\_percent
- 80. linkedin\_headcount\_usa\_percent
- 81. linkedin\_followers\_mom\_percent
- 82. linkedin\_followers\_qoq\_percent
- 83. linkedin\_followers\_yoy\_percent

84. linkedin\_all\_employee\_skill\_names
85. linkedin\_all\_employee\_skill\_count
86. linkedin\_employee\_skills\_0\_to\_10\_pct
87. linkedin\_employee\_skills\_11\_to\_30\_pct
88. linkedin\_employee\_skills\_31\_to\_50\_pct
89. linkedin\_employee\_skills\_51\_to\_70\_pct
90. linkedin\_employee\_skills\_71\_to\_100\_pct
91. glassdoor\_culture\_rating
92. glassdoor\_diversity\_rating
93. glassdoor\_work\_life\_balance\_rating
94. glassdoor\_senior\_management\_rating
95. glassdoor\_compensation\_rating
96. glassdoor\_career\_opportunities\_rating
97. glassdoor\_recommend\_to\_friend\_pct
98. glassdoor\_ceo\_approval\_mom\_pct
99. glassdoor\_ceo\_approval\_qoq\_pct
100. glassdoor\_ceo\_approval\_mom\_pct.1
101. glassdoor\_review\_count\_mom\_pct
102. glassdoor\_review\_count\_qoq\_pct
103. glassdoor\_review\_count\_yoy\_pct
104. g2\_review\_count\_mom\_pct
105. g2\_review\_count\_qoq\_pct
106. g2\_review\_count\_yoy\_pct
107. instagram\_followers (deprecated)
108. instagram\_posts (deprecated)
109. instagram\_followers\_mom\_pct (deprecated)
110. instagram\_followers\_qoq\_pct (deprecated)
111. instagram\_followers\_yoy\_pct (deprecated)

112. recent\_job\_openings\_title  
113. recent\_job\_openings\_title\_count  
114. job\_openings\_count  
115. job\_openings\_count\_mom\_pct  
116. job\_openings\_count\_qoq\_pct  
117. job\_openings\_count\_yoy\_pct  
118. job\_openings\_accounting\_qoq\_pct  
119. job\_openings\_accounting\_six\_months\_growth\_pct  
120. job\_openings\_art\_and\_design\_qoq\_pct  
121. job\_openings\_art\_and\_design\_six\_months\_growth\_pct  
122. job\_openings\_business\_development\_qoq\_pct  
123. job\_openings\_business\_development\_six\_months\_growth\_pct  
124. job\_openings\_engineering\_qoq\_pct  
125. job\_openings\_engineering\_six\_months\_growth\_pct  
126. job\_openings\_finance\_qoq\_pct  
127. job\_openings\_finance\_six\_months\_growth\_pct  
128. job\_openings\_human\_resource\_qoq\_pct  
129. job\_openings\_human\_resource\_six\_months\_growth\_pct  
130. job\_openings\_information\_technology\_qoq\_pct  
131. job\_openings\_information\_technology\_six\_months\_growth\_pct  
132. job\_openings\_marketing\_qoq\_pct  
133. job\_openings\_marketing\_six\_months\_growth\_pct  
134. job\_openings\_media\_and\_communication\_qoq\_pct  
135. job\_openings\_media\_and\_communication\_six\_months\_growth\_pct  
136. job\_openings\_operations\_qoq\_pct  
137. job\_openings\_operations\_six\_months\_growth\_pct  
138. job\_openings\_research\_qoq

## Additional examples

### Crustdata Company Screening API Detailed Examples

# Company Identification API

Given a company's name, website or LinkedIn profile, you can identify the company in Crustdata's database with company identification API

The input to this API is any combination of the following fields

- name of the company
- website of the company
- LinkedIn profile url of the company

## ▼ Request

```
curl 'https://api.crustdata.com/screener/identify/' \ --header 'Accept: application/json, text/plain, */*' \ --header 'Accept-Language: en-US,en;q=0.9' \ --header 'Authorization: Token $api_token' \ --header 'Connection: keep-alive' \ --header 'Content-Type: application/json' \ --header 'Origin: https://crustdata.com' \ --data '{"query_company_website": "serverobotics.com", "count": 1}'
```

Payload fields (at least one of the query fields required):

- `query_company_name` : name of the company
- `query_company_website` : website of the company
- `query_company_linkedin_url` : LinkedIn profile url of the company
- `count` : maximum number of results. Default is 10.

## ▼ Result

Example:

```
[ { "company_id": 628895, "company_name": "Serve Robotics",  
  "company_website_domain": "serverobotics.com", "company_website":  
    "http://www.serverobotics.com", "linkedin_profile_url":  
    "https://www.linkedin.com/company/serverobotics", "linkedin_headcount":  
    82, "acquisition_status": null, "score": 0.3 } ]
```

Each item in the result corresponds to a company record in Crustdata's database. The records are ranked by the matching score, highest first. The score is maximum when all the query fields are provided and their values exactly matches the value of the corresponding company in Crustdata's database.

Each result record contains the following fields for the company

- `company_id` : A unique identifier for the company in Crustdata's database.
- `company_name` : Name of the company in Crustdata's database
- `company_website_domain` : Website domain of the company as mentioned on its Linkedin page
- `company_website` : Website of the company
- `linkedin_profile_url` : LinkedIn profile url for the company
- `linkedin_headcount` : Latest headcount of the company in Crustdata's database
- `acquisition_status` : Either `acquired` or `null`
- `score` : a relative score based on the query parameters provided and how well they match the company fields in Crustdata's database

## Company Dataset API

**Overview:** The Company Dataset API allows users to retrieve specific datasets related to companies, such as job listings, decision makers, news articles, G2 etc.

## ▼ Request Example (Job Listings)

To retrieve data for job listings, make a POST request to the following endpoint:

## Request URL

```
https://api.crustdata.com/data_lab/job_listings/Table/
```

## Request Headers

Header Name	Description	Example Value
Accept	Specifies the types of media that the client can process.	<code>application/json, text/plain, */*</code>
Accept-Language	Specifies the preferred language for the response.	<code>en-US,en;q=0.9</code>
Authorization	Contains the authentication credentials for HTTP authentication.	<code>Token \$token</code>
Content-Type	Indicates the media type of the resource or data.	<code>application/json</code>
User-Agent	Contains information about the user agent (browser) making the request.	<code>Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 ...</code>

## Request Body

Parameter	Type	Description	Example Value
tickers	Array	Can contain specific tickers for filtering.	[ ]
dataset	Object	Contains details about the dataset being requested.	{"name": "job_listings", "id": "joblisting"}
filters	Object	Contains conditions for filtering the data.	See detailed breakdown below.
groups	Array	For grouping the data.	[ ]
aggregations	Array	For data aggregations.	[ ]
functions	Array	For applying functions on the data.	[ ]
offset	Number	The starting point for data retrieval.	0
count	Number	The number of records to retrieve.	100
sorts	Array	For sorting the data.	[ ]

## Filters Object Breakdown

Parameter	Type	Description	Example Value
op	String	The operation for the condition. It can be logical operations like <code>and</code> , <code>or</code> , etc.	and
conditions	Array	An array of conditions. Each condition can have sub-conditions.	See detailed breakdown below.

## Sub-Condition Breakdown

Parameter	Type	Description	Example Value
column	String	The column to be filtered.	company_id
type	String	The type of operation for filtering. Common operations include =, >, <, =>, etc.	=
value	Various	The value for filtering. The datatype can vary based on the column being filtered.	7576

## Response Body

Parameter	Type	Description
fields	Array	An array of objects detailing the attributes of the job listings.
rows	Array	Contains the job listings data. Each entry corresponds to the attributes in the "fields" section.

### Fields Object Breakdown

Parameter	Type	Description
type	String	The data type of the field.
api_name	String	The name used in the API for this field.
hidden	Boolean	Indicates if the field is hidden.
options	Array	Related options for the field.
summary	String	A brief summary of the field.
local_metric	Boolean	Indicates if the field is a local metric.
display_name	String	The display name of the field.
geocode	Boolean	Indicates if the field contains geocode data.

## All dataset endpoints

 [Crustdata Dataset API Detailed Examples](#)

## Search: LinkedIn Company Search API (real-time)

**Overview:** Search for company profiles using either directly a LinkedIn Sales Navigator accounts search URL or a custom search criteria as a filter. This endpoint allows you to retrieve detailed information about companies matching specific criteria.

Each request returns up-to 25 results. To paginate, update the page number of the Sales Navigator search URL and do the request again.

In the request payload, either set the url of the Sales Navigator Accounts search from your browser in the parameter `linkedin_sales_navigator_search_url` or specify the search criteria as a JSON object in the parameter `filters`

Required: authentication token `auth_token` for authorization.

## Building the Company/People Search Criteria Filter

Based on the field on you are filtering, filters can be categorized into 3 different categories

#### ▼ Text Filter

A **text filter** is used to filter based on specific text values. Each **text filter** must contain **filter\_type**, **type** and list of **value**.

Example:

```
{ "filter_type": "COMPANY_HEADCOUNT", "type": "in", "value": ["10,001+", "1,001-5,000"] }
```

Valid **type**:

- **in** : To include values.
- **not in** : To exclude values. Excluding values might not be supported for every filter.

#### ▼ Range Filter

A **range filter** is used to filter based on a range of values. Each filter must contain **filter\_type**, **type** and **value**. Few range filters might contain a **sub\_filter**. Ensure that you correctly pass **sub\_filter** if required.

**sub\_filter**

The **sub\_filter** is an optional field that provides additional context for the range filter. For example, with the **DEPARTMENT\_HEADCOUNT** filter, the **sub\_filter** specifies which department the filter applies to. Ensure that you correctly pass **sub\_filter** if required.

Example:

```
{ "filter_type": "ANNUAL_REVENUE", "type": "between", "value": {"min": 1, "max": 500}, "sub_filter": "USD" }
```

Valid **type**:

- **between** : To specify a range of values, indicating that the value must fall within the defined minimum and maximum limits.

## ▼ Boolean Filter

A **boolean filter** is used to filter based on true/false values. It doesn't contain any **type** or **value**

Example:

```
{ "filter_type": "IN_THE_NEWS" }
```

And here is the full dictionary for filter attributes and possible values you can pass:

## ▼ Filter Dictionary for Company Search

Filter Type	Description	Properties	Value/Sub-filter
COMPANY_HEADCOUNT	Specifies the size of the company based on the number of employees.	types: [in]	"1-10" , "11-50" , "51-200" , "201-500" , "501-1,000" , "1,001-5,000" , "5,001-10,000" , "10,001+"
REGION	Specifies the geographical region of the company.	types: [in, not in]	region_values
INDUSTRY	Specifies the industry of the company.	types: [in, not in]	industry_values
NUM_OF_FOLLOWERS	Specifies the number of followers a company has.	types: [in]	"1-50" , "51-100" , "101-1000" , "1001-5000" , "5001+"
FORTUNE	Specifies the Fortune ranking of the company.	types: [in]	"Fortune 50" , "Fortune 51-100" , "Fortune 101-250" , "Fortune 251-500"
ACCOUNT_ACTIVITIES	Specifies recent account activities, such as leadership changes or funding events.	types: [in]	"Senior leadership changes in last 3 months" , "Funding events in past 12 months"
JOB_OPPORTUNITIES	Specifies job opportunities available at the company.	types: [in]	"Hiring on LinkedIn"
COMPANY_HEADCOUNT_GROWTH	Specifies the growth of the company's headcount.	allowed_without_sub_filter , types: [between]	N/A
ANNUAL_REVENUE	Specifies the annual revenue of the company.	types: [between]	"USD" , "AED" , "AUD" , "BRL" , "CAD" , "CNY" , "DKK" , "EUR" , "GBP" , "HKD"

			<ul style="list-style-type: none"> <li>"IDR" , "ILS" ,</li> <li>"INR" , "JPY" ,</li> <li>"NOK" , "NZD" ,</li> <li>"RUB" , "SEK" ,</li> <li>"SGD" , "THB" ,</li> <li>"TRY" , "TWD"</li> </ul>
DEPARTMENT_HEADCO UNT	Specifies the headcount of specific departments within the company.	types: [between]	<ul style="list-style-type: none"> <li>"Accounting" ,</li> <li>"Administrative" ,</li> <li>"Arts and Design" ,</li> <li>"Business Development" ,</li> <li>"Community and Social Services" ,</li> <li>"Consulting" ,</li> <li>"Education" ,</li> <li>"Engineering" ,</li> <li>"Entrepreneurship" ,</li> <li>"Finance" ,</li> <li>"Healthcare Services" ,</li> <li>"Human Resources" ,</li> <li>"Information Technology" ,</li> <li>"Legal" ,</li> <li>"Marketing" ,</li> <li>"Media and Communication" ,</li> <li>"Military and Protective Services" ,</li> <li>"Operations" ,</li> <li>"Product Management" ,</li> <li>"Program and Project Management" ,</li> <li>"Purchasing" ,</li> <li>"Quality Assurance" ,</li> <li>"Real Estate" ,</li> <li>"Research" ,</li> <li>"Sales" ,</li> <li>"Customer Success and Support"</li> </ul>

<code>DEPARTMENT_HEADCOUNT_GROWTH</code>	Specifies the growth of headcount in specific departments.	<code>types: [between]</code>	<code>"Accounting", "Administrative", "Arts and Design", "Business Development", "Community and Social Services", "Consulting", "Education", "Engineering", "Entrepreneurship", "Finance", "Healthcare Services", "Human Resources", "Information Technology", "Legal", "Marketing", "Media and Communication", "Military and Protective Services", "Operations", "Product Management", "Program and Project Management", "Purchasing", "Quality Assurance", "Real Estate", "Research", "Sales", "Customer Success and Support"</code>
<code>KEYWORD</code>	Filters based on specific keywords related to the company.	<code>types: [in]</code>	List of strings (max length 1)  Supports boolean filters.

Example: `"sales"`  
or `'marketing'` or  
`'gtm'"` will match  
either of these 3  
words across the full  
LinkedIn profile of  
the company

▼ Filter Dictionary for Person Search

Filter Type	Description	Properties	Value/Sub-filter
CURRENT_COMPANY	Specifies the current company of the person.	types: [in, not in]	<p>List of strings.</p> <p>You can specify names, domains or LinkedIn url of the companies.</p> <p>Example:</p> <pre>"Serve Robotics", "serverobotics.com", "https://www.linkedin.com/company/serverobtics"</pre>
CURRENT_TITLE	Specifies the current title of the person.	types: [in, not in]	<p>List of strings. Case insensitive contains matching for each of the strings.</p> <p>Example: ["ceo", "founder", "director"] will match all the profiles with any current job title(s) having any of the 3 strings ("ceo" or "founder" or "director")</p>
PAST_TITLE	Specifies the past titles held by the person.	types: [in, not in]	<p>List of strings. Case insensitive contains matching for each of the strings.</p> <p>Example: ["ceo", "founder", "director"] will match all the profiles with any past job title(s) having any of the 3 strings ("ceo" or "founder" or "director")</p>
COMPANY_HEADQUARTERS	Specifies the headquarters of the	types: [in, not in]	region_values

	person's company.		
COMPANY_HEADCOUNT	Specifies the size of the company based on the number of employees.	types: [in]	"Self-employed", "1-10", "11-50", "51-200", "201-500", "501-1,000", "1,001-5,000", "5,001-10,000", "10,001+"
REGION	Specifies the geographical region of the person.	types: [in, not in]	region_values
INDUSTRY	Specifies the industry of the person's company.	types: [in, not in]	industry_values
PROFILE_LANGUAGE	Specifies the language of the person's profile.	types: [in]	"Arabic", "English", "Spanish", "Portuguese", "Chinese", "French", "Italian", "Russian", "German", "Dutch", "Turkish", "Tagalog", "Polish", "Korean", "Japanese", "Malay", "Norwegian", "Danish", "Romanian", "Swedish", "Bahasa Indonesia", "Czech"
SENIORITY_LEVEL	Specifies the seniority level of the person.	types: [in, not in]	"Owner / Partner", "CXO", "Vice President", "Director", "Experienced Manager", "Entry Level Manager", "Strategic", "Senior", "Entry Level", "In Training"
YEARS_AT_CURRENT_COMP ANY	Specifies the number of years the person has	types: [in]	"Less than 1 year", "1 to 2 years", "3 to 5 years", "5 to 10 years", "More than 10 years"

	been at their current company.		<code>5 years", "6 to 10 years", "More than 10 years"</code>
<code>YEARS_IN_CURRENT_POSITION</code>	Specifies the number of years the person has been in their current position.	<code>types: [in]</code>	<code>"Less than 1 year", "1 to 2 years", "3 to 5 years", "6 to 10 years", "More than 10 years"</code>
<code>YEARS_OF_EXPERIENCE</code>	Specifies the total years of experience the person has.	<code>types: [in]</code>	<code>"Less than 1 year", "1 to 2 years", "3 to 5 years", "6 to 10 years", "More than 10 years"</code>
<code>FIRST_NAME</code>	Specifies the first name of the person.	<code>types: [in]</code>	List of strings (max length 1)
<code>LAST_NAME</code>	Specifies the last name of the person.	<code>types: [in]</code>	List of strings (max length 1)
<code>FUNCTION</code>	Specifies the function or role of the person.	<code>types: [in, not in]</code>	<code>"Accounting", "Administrative", "Arts and Design", "Business Development", "Community and Social Services", "Consulting", "Education", "Engineering", "Entrepreneurship", "Finance", "Healthcare Services", "Human Resources", "Information Technology", "Legal", "Marketing", "Media and Communication", "Military and Protective Services", "Operations", "Product Management", "Program and Project"</code>

			Management" , "Purchasing" , "Quality Assurance" , "Real Estate" , "Research" , "Sales" , "Customer Success and Support"
PAST_COMPANY	Specifies the past companies the person has worked for.	types: [in, not in]	List of strings  You can specify names, domains or LinkedIn url of the companies. Example:  "Serve Robotics" , "serverobotics.com" , "https://www.linkedin.com/company/serverobtics"
COMPANY_TYPE	Specifies the type of company the person works for.	types: [in]	"Public Company" , "Privately Held" , "Non Profit" , "Educational Institution" , "Partnership" , "Self Employed" , "Self Owned" , "Government Agency"
POSTED_ON_LINKEDIN	Specifies if the person has posted on LinkedIn.	N/A	N/A
RECENTLY_CHANGED_JOBS	Specifies if the person has recently changed jobs.	N/A	N/A
IN_THE_NEWS	Specifies if the person has been mentioned in the news.	N/A	N/A
KEYWORD	Filters based on specific keywords related to the company.	types: [in]	List of strings (max length 1)  Supports boolean

## Making Requests

filters.

Example: `"sales" or  
'gtm' or 'marketer'"`  
will match either of  
these 3 words across  
the full LinkedIn profile  
of the person

## ▼ Request:

### Request Body:

The request body can have the following keys (atleast one of them is required)

- `linkedin_sales_navigator_search_url` (optional): URL of the Sales Navigator Accounts search from the browser
- `filters` (optional): JSON dictionary defining the search criteria as laid out by the [Crustdata filter schema](#).
- `page` (optiona): Only valid when `filters` is not empty. When passing `linkedin_sales_navigator_search_url`, page should be specified in `linkedin_sales_navigator_search_url` itself

## Examples

### ▼ Via LinkedIn Sales Navigator URL:

```
curl --location 'https://api.crustdata.com/screener/company/search' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json, text/plain, */*' \
--header 'Accept-Language: en-US,en;q=0.9' \
--header 'Authorization: Token $auth_token' \
--data '{ "linkedin_sales_navigator_search_url": "https://www.linkedin.com/sales/search/company?query=(filters%3AList((type%3ACOMPANY_HEADCOUNT%2Cvalues%3AList((id%3AD%2Ctext%3A51-200%2CselectionType%3AINCLUDED)))%2C(type%3AREGION%2Cvalues%3AList((id%3A103323778%2Ctext%3AMexico%2CselectionType%3AINCLUDED)))%2C(type%3AINDEX%2Cvalues%3AList((id%3A25%2Ctext%3AManufacturing%2CselectionType%3AINCLUDED))))&sessionId=8TR8HMz%2BTVOYaeivK9p%2Bpg%3D%3D&viewAllFilters=true" }'
```

### Via Custom Search Filters:

Refer [Building the Company/People Search Criteria Filter](#) to build the custom search filter for your query and pass it in the `filters` key. Each element of `filters` is a JSON object which defines a filter on a specific field. All the elements of `filters` are joined with a logical “AND” operation when doing the search.

Example:

This query retrieves people from companies with a headcount between **1,001-5,000** or more than **10,001+**, with annual revenue between **1** and **500 million USD**, excluding those located in the **United States**, and returns the second page of results.

```
curl --location 'https://api.crustdata.com/screener/company/search' \ --  
header 'Content-Type: application/json' \ --header 'Accept:  
application/json, text/plain, */*' \ --header 'Accept-Language: en-US,en;q=0.9' \ --header 'Authorization: Token $token' \ --data '{  
"filters": [ { "filter_type": "COMPANY_HEADCOUNT", "type": "in", "value":  
["10,001+", "1,001-5,000"] }, { "filter_type": "ANNUAL_REVENUE", "type":  
"between", "value": {"min": 1, "max": 500}, "sub_filter": "USD" }, {  
"filter_type": "REGION", "type": "not in", "value": ["United States"] }  
], "page": 2 }'
```

#### ▼ Response:

 json\_hero JSON Hero

## ▼ Key points:

- **Credits:** Each page request costs 25 credits
- **Pagination:** If the total number of results for the query is more than 25 (value of `total_display_count` param), you can paginate the response in the following ways (depending on your request)
  - When passing `linkedin_sales_navigator_search_url` :
    - adding `page` query param to `linkedin_sales_navigator_search_url` . For example, to get data on `n` th page, `linkedin_sales_navigator_search_url` would become `https://www.linkedin.com/sales/search/company?page=n&query=...` .
      - ▶ Example request with `page=2`
  - When passing `filters` :
    - provide `page` as one of the keys in the payload itself
      - ▶ Example request with `page=2`

Each page returns upto 25 results. To fetch all the results from a query, you should keep on iterating over pages until you cover the value of `total_display_count` in the response from first page.

- **Latency:** The data is fetched in real-time from LinkedIn and the latency for this endpoint is between 10 to 30 seconds.
- **Response schema:** Because the data is fetched realtime, and the results may not be in Crustdata's database already, the response schema will be different from company data enrichment endpoint `screener/company` . But all the results will be added to Crustdata's database in 60 min of your query and the data for a specific company profile can be enriched via company enrichment endpoint

## LinkedIn Posts by Company API (real-time)

**Overview:** This endpoint retrieves recent LinkedIn posts and related engagement metrics for a specified company.

Each request returns up-to 5 results per page. To paginate, increment the `page` query param.

Required: authentication token `auth_token` for authorization.

## ▼ Request

- **Use Case:** Ideal for users who want to fetch recent LinkedIn posts and engagement data for a specific company.
- **Note:** You can provide one company LinkedIn URL per request.
- Request Parameters:
  - `company_name` (optional): Company name
  - `company_domain` (optional): Company domain
  - `company_id` (optional): Company ID
  - `company_linkedin_url` (optional): Company LinkedIn URL
  - ▶ `fields` (optional): comma separated list of fields which you want to get in response.
  - `page` (optional, default: 1): Page number for pagination
  - `limit` (optional, default: 5): Limit the number of posts in a page
  - ▶ `post_types` (optional, default: repost, original)

**Note:** Provide only one of the company identifiers.

- **Example Request:**

- ▶ With default `fields`
- ▶ With default `fields` + reactors
- ▶ With default `post_types`

## ▼ Response

The response provides a list of recent LinkedIn posts for the specified company, including post content, engagement metrics, and information about users who interacted with the posts.

Full sample:  [jsonhero.io](#)

▼ Response Structure:

```
{ "posts": [ { "backend_urn": "urn:li:activity:7236812027275419648", "share_urn": "urn:li:share:7236812026038083584", "share_url": "https://www.linkedin.com/posts/crustdata_y-combinators-most-popular-startups-from-activity-7236812027275419648-4fyw?utm_source=combined_share_message&utm_medium=member_desktop", "text": "Y Combinator's most popular startups.\nFrom the current S24 batch.\nHow do you gauge the buzz around these startups when most are pre-product?\nWe've defined web traffic as the metric to go by.\nHere are the most popular startups from YC S24: \n\n1. NextUI: Founded by Junior Garcia\n2. Wordware: Filip Kozera, Robert Chandler\n3. Unriddle: Naveed Janmohamed\n4. Undermind: Thomas Hartke, Joshua Ramette\n5. Comfydeploy: Nick Kao, Benny Kok\n6. Beebettor: Jordan Murphy, Matthew Wolfe\n7. Merse: Kumar A., Mark Rachapoom\n8. Laminar: Robert Kim, Din Mailibay, Temirlan Myrzakhmetov\n9. MitoHealth: Kenneth Lou, Tee-Ming C., Joel Kek, Ryan Ware\n10. Autarc: Etienne-Noel Krause, Thies Hansen, Marius Seufzer\n\nInterested in reading more about the YC S24 batch? Read our full breakdown from the link in the comments ↗", "actor_name": "Crustdata", "date_posted": "2024-09-03", "hyperlinks": { "company_linkedin_urls": [], "person_linkedin_urls": [ "https://www.linkedin.com/in/ACoAAAKoldoBqSsiXY_DHsXdSk1slibabeTvDDY" ], "other_urls": [] }, "total_reactions": 37, "total_comments": 7, "reactions_by_type": { "LIKE": 28, "EMPATHY": 4, "PRAISE": 4, "INTEREST": 1 }, "num_shares": 5, "is_repost_without_thoughts": false, "reactors": [ { "name": "Courtney May", "linkedin_profile_url": "https://www.linkedin.com/in/ACwAACkMyzkBYncrCuM2rzhc06iz6oj741NL-98", "reaction_type": "LIKE", "profile_image_url": "https://media.licdn.com/dms/image/v2/D5603AQF-8vL_c5H9Zg/profile-displayphoto-shrink_100_100/profile-displayphoto-shrink_100_100/0/1690558480623?e=1730937600&v=beta&t=Lm2hHLTFiEVlHWdTt-Vh3vDYevK8U8SlPqaFdNu3R6A", "title": "GTM @ Arc (YC W22)", "additional_info": "3rd+", "location": "San Francisco, California, United States", "linkedin_profile_urn": "ACwAACkMyzkBYncrCuM2rzhc06iz6oj741NL-98", "default_position_title": "GTM @ Arc (YC W22)", "default_position_company_linkedin_id": "74725230", "default_position_is_decision_maker": false, "flagship_profile_url": "https://www.linkedin.com/in/courtney-may-8a178b172", "profile_picture_url": "https://media.licdn.com/dms/image/v2/D5603AQF-8vL_c5H9Zg/profile-displayphoto-shrink_400_400/profile-displayphoto-shrink_400_400/0/1690558480623?" } ] }
```

```
e=1730937600&v=beta&t=vHg233746zA00m3q2vHKSFcthL3YKiaqTtVEZt1qqJI",
"headline": "GTM @ Arc (YC W22)", "summary": null,
"num_of_connections": 786, "related_colleague_company_id": 74725230,
"skills": [ "Marketing Strategy", "Product Support", "SOC 2", ... ],
"employer": [ { "title": "GTM @ Arc (YC W22)", "company_name": "Arc",
"company_linkedin_id": "74725230", "start_date": "2024-07-
01T00:00:00", "end_date": null, "description": null, "location": "San
Francisco, California, United States", "rich_media": [] }, { "title":
"Product Marketing & Operations Lead", "company_name": "Bits of
Stock™", "company_linkedin_id": "10550545", "start_date": "2023-03-
01T00:00:00", "end_date": "2024-07-01T00:00:00", "description": "•
Spearheaded SOC 2 Certification and oversaw compliance organization
for internal and external needs.\n• Leads a weekly operations call to
manage customer support, new user onboarding, and other outstanding
operational matters.\n• Wrote & launched: Product Blog with 6
different featured pieces; 2 Pricing Thought-Leadership pieces; & 2
Partner Press Releases; two of which were featured in the WSJ.\n•
Managed marketing and logistics for 11 conferences and events all
over the world, producing over 150 B2B qualified leads.\n• Created a
company-wide marketing strategy and implemented it across the blog,
LinkedIn, & Twitter leading to a 125% increased engagement rate & a
29% increase in followers.\n• Aided in sales and partner relations by
preparing a Partner Marketing Guide, creating the user support
section of the website and inbound email system, and investing
education guide.", "location": "San Francisco Bay Area",
"rich_media": [] }, ... ], "education_background": [ { "degree_name":
"Bachelor of Applied Science - BASc", "institute_name": "Texas
Christian University", "field_of_study": "Economics", "start_date":
"2016-01-01T00:00:00", "end_date": "2020-01-01T00:00:00" } ],
"emails": [ "email@example.com" ], "websites": [], "twitter_handle":
null, "languages": [], "pronoun": null, "current_title": "GTM @ Arc
(YC W22)" }, ... ] } ] }
```

Each item in the `posts` array contains the following fields:

- `backend_urn` (string): Unique identifier for the post in LinkedIn's backend system.
- `share_urn` (string): Unique identifier for the shared content.
- `share_url` (string): Direct URL to the post on LinkedIn.
- `text` (string): The full content of the post.
- `actor_name` (string): Name of the company or person who created the post.

- `hyperlinks` (object): Contains the external links and Company/Person LinkedIn urls mentioned in the post
  - `company_linkedin_urls` (array): List of Company LinkedIn urls mentioned in the post
  - `person_linkedin_urls` (array): List of Person LinkedIn urls mentioned in the post
- `date_posted` (string): Date when the post was published, in "YYYY-MM-DD" format.
- `total_reactions` (integer): Total number of reactions on the post.
- `total_comments` (integer): Total number of comments on the post.
- `reactions_by_type` (object): Breakdown of reactions by type.
  - Possible types include: "LIKE", "EMPATHY", "PRAISE", "INTEREST", etc.
  - Each type is represented by its count (integer).
- `num_shares` (integer): Number of times the post has been shared.

- **reactors** (array): List of users who reacted to the post. Each reactor object contains:
  - **name** (string): Full name of the person who reacted.
  - **linkedin\_profile\_url** (string): URL to the reactor's LinkedIn profile.
  - **reaction\_type** (string): Type of reaction given (e.g., "LIKE", "EMPATHY").
  - **profile\_image\_url** (string): URL to the reactor's profile image (100x100 size).
  - **title** (string): Current professional title of the reactor.
  - **additional\_info** (string): Additional information, often indicating connection degree.
  - **location** (string): Geographic location of the reactor.
- ▼ **Key Points**
  - **linkedin\_profile\_urn** (string): Unique identifier for the reactor's LinkedIn profile.
- **Credits:**
  - **default\_position\_title** (string): Primary job title.
  - Without reactors (default): Each successful page request costs 5 credits
    - **default\_position\_company\_linkedin\_id** (string): LinkedIn ID of the reactor's primary company.
  - With reactors: Each successful page request costs 25 credits
- **Pagination:**
  - **default\_position\_is\_decision\_maker** (boolean): Indicates if the reactor is in a decision-making role.
  - Increment the value of **page** query param to fetch the next set of posts.
  - Most **recent\_posts\_will\_be\_in** (string): Another form of the reactor's LinkedIn profile URL.
  - Currently, you can only fetch only upto 20 pages of latest posts. In case you want to fetch more, contact Crustdata team for a larger version of the profile picture (400x400 size).
- External urls or Company/Person LinkedIn urls mentioned in text:
  - **headline** (string): Professional headline from the reactor's LinkedIn profile.
  - **hyperlinks** contains list of links (categorized as **company\_linkedin\_urls**, **person\_linkedin\_urls** and **other\_urls**) mentioned in the post
- **Latency:** The data is fetched in real-time from LinkedIn and the latency for this endpoint is between 30 to 60 seconds. Number of LinkedIn connections for all the posts in the page
  - **related\_colleague\_company\_id** (integer): LinkedIn ID of a related

## LinkedIn Posts Keyword Search (real-time)

- **skills** (array of strings): List of professional skills listed on the reactor's profile.

**Overview:** This endpoint retrieves linked posts containing the specified key word along with related engagement metrics.

▪ `title` (string): Job title.

- `company_name` (string): Name of the employer.

Each request returns 5 posts per page. To paginate, increment the `page` in the payload.

▪ `company_linkedin_id` (string or null): LinkedIn ID of the company.

- `start_date` (string): Start date of employment in ISO format.

▪ `end_date` (string or null): End date of employment in ISO format, or null if current.

- `description` (string or null): Job description, if available.

- `location` (string or null): Job location.

- `rich_media` (array): Currently empty, may contain media related to the job.

- `education_background` (array of objects): Educational history, each containing:

- `degree_name` (string): Type of degree obtained.

- `institute_name` (string): Name of the educational institution.

- `field_of_study` (string): Area of study.

- `start_date` (string): Start date of education in ISO format.

- `end_date` (string): End date of education in ISO format.

- `emails` (array of strings): Known email addresses associated with the reactor.

- `websites` (array): Currently empty, may contain personal or professional websites.

- `twitter_handle` (string or null): Twitter username, if available.

- `languages` (array): Currently empty, may contain languages spoken.

- `pronoun` (string or null): Preferred pronouns, if specified.

- `current_title` (string): Current job title, often identical to `default_position_title`.

## ▼ Request

### Request Body Overview

The request body is a JSON object that contains the following parameters:

Parameter	Description
keyword	The keyword or phrase to search for in LinkedIn posts.
page	Page number for pagination
limit	Limit the number of posts in a page
sort_by	Defines the sorting order of the results Can be either of the following: 1. "relevance" - to sort on top match 2. "date_posted" - to sort on latest posts
date_posted	Filters posts by the date they were posted. Can be one of the following: 1. "past-24h" - Posts from last 24 hours 2. "past-week" - Post from last 7 days 3. "past-month" - Post from last 30 days 4. "past-quarter" - Post from last 3 months 5. "past-year" - Post from last 1 year

\* `limit` can not exceed 5 when `page` is provided in the payload. To retrieve posts in bulk, use the `limit` parameter (with value over 5 allowed here) without the `page` parameter.

In the example below, we get LinkedIn posts that meet the following criteria:

- Get all the posts with "*LLM evaluation*" keyword
- Posted in last 3 months

## ▼ cURL

```
curl  
'https://api.crustdata.com/screener/linkedin_posts/keyword_search/' \  
-H 'Accept: application/json, text/plain, /' \ -H 'Accept-Language:  
en-US,en;q=0.9' \ -H 'Authorization: Token $auth_token' \ -H  
'Connection: keep-alive' \ -H 'Content-Type: application/json' \ -H  
'Origin: https://crustdata.com' \ --data-raw '{ "keyword": "LLM  
Evaluation", "page": 1, "sort_by": "relevance", "date_posted": "past-  
quarter" }' \ --compressed
```

## ▼ Python

```
import requests  
headers = { 'Accept': 'application/json, text/plain, /',  
'Language': 'en-US,en;q=0.9', 'Authorization': 'Token $auth_token', # rep  
$auth_token 'Connection': 'keep-alive', 'Content-Type': 'application/jsc  
'Origin': 'https://crustdata.com' }  
json_data = { "keyword": "LLM Evaluation",  
"page": 1, "sort_by": "relevance", "date_posted": "past-quarter" }  
response = requests.post('https://api.crustdata.com/screener/linkedin_posts/keyword_search',  
headers=headers, json=json_data)
```



## ▼ Response:

The response provides a list of recent LinkedIn posts for the specified company, including post content, engagement metrics, and information about users who interacted with the posts.

Refer to `actor_type` field to identify if the post is published by a person or a company

Full sample: <https://jsonhero.io/j/XlqoVuhe2x9w>

## ▼ Key Points

- **Credits:**
  - Each successful page request costs 5 credits.
- **Pagination:**
  - Increment the value of `page` query param to fetch the next set of posts.  
Each page has 5 posts.
  - `limit` can not exceed 5 when `page` is provided in the payload. To retrieve posts in bulk, use the `limit` parameter (with value over 5 allowed here) without the `page` parameter.
- **Latency:** The data is fetched in real-time from LinkedIn and the latency for this endpoint is between 5 to 10 seconds depending on number of posts fetched in a request.

# People Endpoints

## Enrichment: People Profile(s) API

**Overview:** Enrich data for one or more individuals using LinkedIn profile URLs or business email addresses. This API allows you to retrieve enriched person data from Crustdata's database or perform a real-time search from the web if the data is not available.

### Key Features:

- Enrich data using **LinkedIn profile URLs** or **business email addresses** (3 credit per profile/email)
- Option to perform a **real-time search** if data is not present in the database (5 credit per profile/email)
- Retrieve data for up to **25 profiles or emails** in a single request.

Required: authentication token `auth_token` for authorization.



## ▼ Request:

### Query Parameters

- ***linkedin\_profile\_url*** (optional): Comma-separated list of LinkedIn profile URLs.

- **Example:**

```
linkedin_profile_url=https://www.linkedin.com/in/johndoe/,https://www.  
linkedin.com/in/janedoe/
```

```
curl 'https://api.crustdata.com/screener/person/enrich?  
linkedin_profile_url=https://www.linkedin.com/in/dtpow/,https://www.  
\ --header 'Accept: application/json, text/plain, */*' \ --header 'A  
US,en;q=0.9' \ --header 'Authorization: Token $auth_token'
```

- ***business\_email*** (optional): Person business email address.

- **Note:-** You can only provide one business email address per request

- **Example:** `business_email=john.doe@example.com`

```
curl 'https://api.crustdata.com/screener/person/enrich?  
business_email=john.doe@example.com' \ --header 'Accept:  
application/json, text/plain, */*' \ --header 'Accept-Language:  
en-US,en;q=0.9' \ --header 'Authorization: Token $auth_token'
```

- ***enrich\_realtime*** (optional): Boolean (True or False). If set to True, performs a real-time search from the web if data is not found in the database.

- **Default:** False

- **Example:**

```
curl 'https://api.crustdata.com/screener/person/enrich?  
linkedin_profile_url=https://www.linkedin.com/in/dtpow/,https://www.  
\ --header 'Accept: application/json, text/plain, */*' \ --header 'A  
'Authorization: Token $auth_token'
```

- **fields** (optional): *string* (comma-separated list of fields). Specifies the fields you want to include in the response.

▼ Possible Values

- linkedin\_profile\_url: *string*
- linkedin\_flagship\_url: *string*