Crustdata Discovery And Enrichment API

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.

Getting Started

Obtaining Authorization Token

Reach out to 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://linkedin.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="HubSpot","Google>,
Inc."' \\
  --header 'Accept: application/json, text/plain, */*' \\
  --header 'Accept-Language: en-US,en;q=0.9' \\
  --header '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://linkedin.com/company/clay-hq>' \\
  --header 'Accept: application/json, text/plain, */*' \\
  --header '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_headcount>' \\
  --header 'Authorization: Token $token' \\
  --header 'Accept: application/json'
```

-

More examples of Using `fields` parameter

Example 1: Request Specific Top-Level Fields

Request:

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

- **Response Includes:**
  - **company_name**
  - **company_website_domain**
  - rest of top-level fields

Example 2: Request Nested Fields

Request:

```
 curl
'<https://api.crustdata.com/screener/company?company_id=123&fields=glassdoor.glassdoor_overall_rating,glassdoor.glassdoor_review_count>' \\
  --header 'Authorization: Token $token' \\
  --header 'Accept: application/json'
```

- **Response Includes:**
  - **glassdoor**
    - **glassdoor_overall_rating**
    - **glassdoor_review_count**
  - rest of top-level fields

Example 3: Include 'decision_makers' and 'founders.profiles'

Request:

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

- **Response Includes:**
    - **decision_makers**: Full array of decision-maker profiles.
    - **founders**
        - **profiles**: Full array of founder profiles.
    - rest of top-level fields

Example 4: Requesting Unauthorized Fields

Assuming you do not have access to the `headcount` field.

Request:

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

Error Response:

```
 {
  "error": "Unauthorized access to field(s): headcount"
}
```

-
- **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_realtime=True>' \\
--header 'Accept: application/json, text/plain, /' \\
--header 'Accept-Language: en-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_specialities**: *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 https://jsonhero.io/j/QN8Qj7dg8MbW

- **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	Required
metrics	An array of metric objects containing the metric name. Value should always be	
`[{"metric_name": "linkedin_headcount_and_glassdoor_ceo_approval_and_g2"}]`	Yes	
filters	An object containing the filter conditions.	Yes

offset	The starting point of the result set. Default value is 0.	Yes
count	The number of results to return in a single request.	
Maximum value is `100`.		
Default value is `100`.	Yes	
sorts	An array of sorting criteria.	No

○

Parameters:

`metrics`

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

[{"metric_name": "linkedin_headcount_and_glassdoor_ceo_approval_and_g2"}]

1.

`filters`

Example:

```
{
  "op": "and",
  "conditions": [
                 {
                                "op": "or",
                                "conditions": [
                                                     {"hq_country", "type": "(.)", "value":
"USA"},
                                                  {"hq_country", "type": "(.)", "value":
"IND"}
                                          ],
                              }
    {"column": "crunchbase_total_investment_usd", "type": "=>", "value": "5000000"},
    {"column": "largest_headcount_country", "type": "(.)", "value": "USA"}
  ]
}
```

2. The filters object contains the following parameters:

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

3. **`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

[Crustdata Data Dictionary]

- List of all `type` values

condition type	condition description	applicable column types	example
"=>"	Greater than or equal	number	{ "column": "linkedin_headcount", "type": "=>", "value": "50"}
"=<"	Lesser than or equal	number	{ "column": "linkedin_headcount", "type": "=<", "value": "50"}
"=",	Equal	number	{ "column": "linkedin_headcount", "type": "=", "value": "50"}
“<”	Lesser than	number	{ "column": "linkedin_headcount", "type": "<", "value": "50"}
“>”	Greater than	number	{ "column": "linkedin_headcount", "type": ">", "value": "50"}
“(.)”	Contains, case insensitive	string	{ "column": "crunchbase_categories" , "type": "(.)", "value": "artificial intelligence"}
“[.]”	Contains, case sensitive	string	{ "column": "crunchbase_categories" , "type": "[.]", "value": "Artificial Intelligence"}
"!="	Not equals	number	
“in”	Exactly matches atleast one of the elements of list	string, number	{ "column": "company_id", "type": "in", "value": [123, 346. 564]}

Complex Filter Object

Example:

```
{
        "op": "or",
        "conditions": [
                        {"hq_country", "type": "(.)", "value": "USA"},
                        {"hq_country", "type": "(.)", "value": "IND"}
        ]
}
```

2. Same schema as the parent **`filters`** parameter

- **Response**

  Example: https://jsonhero.io/j/ntHvSKVeZJIc

  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:

    1. For each row, iterate over each value.
    2. 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.	`application/json, text/plain, */*`
Accept-Language	Specifies the preferred language for the response.	`en-US,en;q=0.9`
Authorization	Contains the authentication credentials for HTTP authentication.	`Token $token`
Content-Type	Indicates the media type of the resource or data.	`application/json`
User-Agent	Contains information about the user agent (browser) making the request.	`Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 ...`

-

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 `and`, `or`, 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"`, `"IDR"`, `"ILS"`, `"INR"`, `"JPY"`, `"NOK"`, `"NZD"`, `"RUB"`, `"SEK"`, `"SGD"`, `"THB"`, `"TRY"`, `"TWD"`

DEPARTMENT_HEADCOUNT	Specifies the headcount of specific departments within the company.	types: [between]	"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"`
`DEPARTMENT_HEADCOUNT_GROWTH`	Specifies the growth of headcount in specific departments.	`types: [between]`	`"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"`
`KEYWORD`	Filters based on specific keywords related to the company.	`types: [in]`	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]`	List of strings.

- You can specify names, domains or LinkedIn url of the companies. Example:

  `”Serve Robotics”`, `“serverobotics.com”`, `“<https://www.linkedin.com/company/serverobotics”>` | | `CURRENT_TITLE` | Specifies the current title of the person. | `types: [in, not in]` | List of strings. Case in-sensitive contains matching for each of the strings.

  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”) | |

PAST_TITLE | Specifies the past titles held by the person. | types: [in, not in] | List of strings. Case in-sensitive contains matching for each of the strings.

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”) | | COMPANY_HEADQUARTERS | Specifies the headquarters of the person's company. | types: [in, not in] | region_values | | 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_COMPANY | Specifies the number of years the person has been at their current company. | types: [in] | "Less than 1 year", "1 to 2 years", "3 to 5 years", "6 to 10 years", "More than 10 years" | | YEARS_IN_CURRENT_POSITION | Specifies the number of years the person has been in their current position. | types: [in] | "Less than 1 year", "1 to 2 years", "3 to 5 years", "6 to 10 years", "More than 10 years" | | YEARS_OF_EXPERIENCE | Specifies the total years of experience the person has. | types: [in] | "Less than 1 year", "1 to 2 years", "3 to 5 years", "6 to 10 years", "More than 10 years" | | FIRST_NAME | Specifies the first name of the person. | types: [in] | List of strings (max length 1) | | LAST_NAME | Specifies the last name of the person. | types: [in] | List of strings (max length 1) | | FUNCTION | Specifies the function or role of the person. | types: [in, not in] | "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" | | 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/serverobotics”`> | | `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 filters.

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

Making Requests

- **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%3AINDUSTRY%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**:

  https://jsonhero.io/j/zn02zfopXQas

- **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`

```
 curl --location '<https://api.crustdata.com/screener/person/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?page=2&query=(filters%3AList((type%3ACOMPANY_HEADCOUNT%2Cvalues%3AList((id%3AD%2Ctext%3A51-200%2CselectionType%3AINCLUDED)>))%2C(type%3AINDUSTRY%2Cvalues%3AList((id%3A25%2Ctext%3AManufacturing%2CselectionType%3AINCLUDED)))))&sessionId=8TR8HMz%2BTVOYaeivK9p%2Bpg%3D%3D"
}'
```

■

        - When passing `filters` :

            - provide `page` as one of the keys in the payload itsefl

Example request with `page=2`

```
 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
}'
```

            -

    - 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.
      - all possible values:
        - total_reactions
        - total_comments
        - text
        - share_urn
        - share_url
        - reactors
        - reactions_by_type.PRAISE
        - reactions_by_type.LIKE
        - reactions_by_type.INTEREST
        - reactions_by_type.ENTERTAINMENT
        - reactions_by_type.EMPATHY
        - reactions_by_type.CURIOUS

- reactions_by_type.APPRECIATION
  - reactions_by_type
  - num_shares
  - hyperlinks.person_linkedin_urls
  - hyperlinks.other_urls
  - hyperlinks.company_linkedin_urls
  - hyperlinks
  - date_posted
  - backend_urn
  - actor_name
  - year_founded
- default: All fields except `reactors` :`total_reactions,total_comments,text,share_urn,share_url,reactions_by_type_PRAISE,reactions_by_type_LIKE,reactions_by_type_INTEREST,reactions_by_type_ENTERTAINMENT,reactions_by_type_EMPATHY,reactions_by_type_CURIOUS,reactions_by_type_APPRECIATION,reactions_by_type,num_shares,hyperlinks_person_linkedin_urls,hyperlinks_other_urls,hyperlinks_company_linkedin_urls,hyperlinks,date_posted,backend_urn,actor_name,year_founded`
- `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)

  - All post types
    - `original`: only original posts are returned
    - `repost` : only reposted posts are returned

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

- **Example Request:**

With default `fields`

```
 curl
'<https://api.crustdata.com/screener/linkedin_posts?company_domain=https://crustdata.com&page=1>' \\
  --header 'Accept: application/json, text/plain, */*' \\
  --header 'Accept-Language: en-US,en;q=0.9' \\
  --header 'Authorization: Token $auth_token'
```

■

With default `fields` + reactors

```
 curl
'<https://api.crustdata.com/screener/linkedin_posts?company_domain=https://crustdata.com&page=1&fields=reactors>' \\
  --header 'Accept: application/json, text/plain, */*' \\
  --header 'Accept-Language: en-US,en;q=0.9' \\
  --header 'Authorization: Token $auth_token'
```

■

With default `post_types`

```
 curl
'<https://api.crustdata.com/screener/linkedin_posts?company_domain=https://crustdata.com&page=1&post_types=repost%2C%20original>' \\
  --header 'Accept: application/json, text/plain, */*' \\
  --header 'Accept-Language: en-US,en;q=0.9' \\
  --header 'Authorization: Token $auth_token'
```

■

- **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: https://jsonhero.io/j/8O15yo2SVBHD

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.\\n\\nHow do you gauge the buzz around these startups when most are pre-product?\\n\\nWe’ve defined web traffic as the metric to go by.\\n\\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\\n🤔 Interested in reading more about the YC S24 batch?\\n\\nRead 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=vHg233746zA00m3q2vHKSFcthL3YKiagTtVEZt1qqJl>",
      "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.
- `linkedin_profile_urn` (string): Unique identifier for the reactor's LinkedIn profile.
- `default_position_title` (string): Primary job title.
- `default_position_company_linkedin_id` (string): LinkedIn ID of the reactor's primary company.
- `default_position_is_decision_maker` (boolean): Indicates if the reactor is in a decision-making role.
- `flagship_profile_url` (string): Another form of the reactor's LinkedIn profile URL.
- `profile_picture_url` (string): URL to a larger version of the profile picture (400x400 size).
- `headline` (string): Professional headline from the reactor's LinkedIn profile.
- `summary` (string or null): Brief professional summary, if available.
- `num_of_connections` (integer): Number of LinkedIn connections the reactor has.
- `related_colleague_company_id` (integer): LinkedIn ID of a related company, possibly current employer.
- `skills` (array of strings): List of professional skills listed on the reactor's profile.
- `employer` (array of objects): Employment history, each containing:
    - `title` (string): Job title.
    - `company_name` (string): Name of the employer.
    - `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`.
- **Key Points**
    - **Credits:**
        - Without reactors (default): Each successful page request costs 5 credits
        - With reactors: Each successful page request costs 25 credits
    - **Pagination:**
        - Increment the value of `page` query param to fetch the next set of posts.
        - Most recent posts will be in first page and then so on.
        - Currently, you can only fetch only upto 20 pages of latest posts. In case you want to fetch more, contact Crustdata team at info@crustdata.com .
    - External urls or Company/Person LinkedIn urls mentioned in text:
        - `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 depending on number of reactions for all the posts in the page

LinkedIn Posts Keyword Search (real-time)

Overview: This endpoint retrieves LinkedIn posts containing specified keywords along with related engagement metrics.

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

Required: authentication token `auth_token` for authorization.

- **Request**

Request Body Overview

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

Parameter	Description	Default	Required
keyword	The keyword or phrase to search for in LinkedIn posts.		Yes
page	Page number for pagination	1	Yes
limit	Limit the number of posts in a page	5	No
sort_by	Defines the sorting order of the results		
Can be either of the following:			

- "relevance" - to sort on top match
- "date_posted" - to sort on latest posts | "date_posted" | No | | date_posted | Filters posts by the date they were posted. Can be one of the following:
- "past-24h" - Posts from last 24 hours
- "past-week" - Post from last 7 days
- "past-month" - Post from last 30 days
- "past-quarter" - Post from last 3 months
- "past-year" - Post from last 1 year | "past-24h" | No |
- `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, /',
   '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 = {
  "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/XIqoVuhe2x9w

- **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.linkedin.com/in/janedoe/>' \\
  --header 'Accept: application/json, text/plain, */*' \\
  --header 'Accept-Language: en-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.linkedin.com/in/janedoe/&enrich_realtime=True>' \\
  --header 'Accept: application/json, text/plain, */*' \\
  --header 'Accept-Language: en-US,en;q=0.9' \\
  --header '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*
        - name: *string*
        - location: *string*
        - email: *string*
        - title: *string*
        - last_updated: *string*
        - headline: *string*
        - summary: *string*
        - num_of_connections: *string*
        - skills: *array of strings*
        - profile_picture_url: *string*
        - twitter_handle: *string*
        - languages: *array of strings*
        - linkedin_open_to_cards: *array of strings*
        - all_employers: *array of objects*
        - past_employers: *array of objects*
        - current_employers: *array of objects*
        - education_background.degree_name: key with string value in array of objects
        - education_background.end_date: key with string value in array of objects
        - education_background.field_of_study: key with string value in array of objects
        - education_background.institute_linkedin_id: key with string value in array of objects
        - education_background.institute_linkedin_url: key with string value in array of objects
        - education_background.institute_logo_url: key with string value in array of objects
        - education_background.institute_name: key with string value in array of objects
        - education_background.start_date: key with string value in array of objects
        - education_background.activities_and_societies: key with string value in array of objects
        - certifications: *array of objects*
        - honors: *array of objects*
        - all_employers_company_id: *array of integers*
        - all_titles: *array of strings*
        - all_schools: *array of strings*
        - all_degrees: *array of strings*
        - all_connections: *array of strings*

        - **Example:** `fields=all_degrees,education_background`
- **Notes:**
    - **Mandatory Parameters:** You must provide either `linkedin_profile_url` or `business_email`. Do not include both in the same request.
    - **Formatting:** Multiple URLs or emails should be comma-separated. Extra spaces or tabs before or after commas are ignored.
    - Multiple LinkedIn profile URLs should be separated by commas. Extra spaces or tabs before or after commas will be ignored.
    - **Fields**
        - If you don't use fields, you will get all the fields in response except `all_connections`, `linkedin_open_to_cards`,`certifications` , `honors` & `education_background.activities_and_societies`
        - 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.
        - Top level non-object fields are present in response irrespective of fields.
        - Don't include metadata fields : `enriched_realtime`, `score` and `query_linkedin_profile_urn_or_slug` in fields
- **Examples**
    - **1. Request with all fields**:
        - Usecase: Ideal for users who wants to access all fields which are not provided by default

```
curl -X GET
"<https://api.crustdata.com/screener/person/enrich?linkedin_profile_url=https://www.linkedin.com/in/sasikumarm00&enrich_realtime=true&fields=linkedin_profile_url,linkedin_flagship_url,name,location,email,title,last_updated,headline,summary,num_of_connections,skills,profile_picture_url,twitter_handle,languages,linkedin_open_to_cards,all_employers,past_employers,current_employers,education_background.degree_name,education_background.end_date,education_background.field_of_study,education_background.institute_linkedin_id,education_background.institute_linkedin_url,education_background.institute_logo_url,education_background.institute_name,education_background.start_date,education_background.activities_and_societies,certifications,honors,all_employers_company_id,all_titles,all_schools,all_degrees,all_connections>" \\
-H "Authorization: Token auth_token" \\,
-H "Content-Type: application/json"
```

○

2. Request with all default fields AND `education_background.activities_and_societies`:

```
 curl -X GET
"<https://api.crustdata.com/screener/person/enrich?linkedin_profile_url=https://www.linkedin.com/in/sasikumarm00&enrich_realtime=true&fields=linkedin_profile_url,linkedin_flagship_url,name,location,email,title,last_updated,headline,summary,num_of_connections,skills,profile_picture_url,twitter_handle,languages,all_employers,past_employers,current_employers,education_background.degree_name,education_background.end_date,education_background.field_of_study,education_background.institute_linkedin_id,education_background.institute_linkedin_url,education_background.institute_logo_url,education_background.institute_name,education_background.start_date,education_background.activities_and_societies,all_employers_company_id,all_titles,all_schools,all_degrees>" \\
-H "Authorization: Token auth_token" \\
-H "Content-Type: application/json"
```

○

3. Request with all default fields AND `certifications` , `honors` and `linkedin_open_to_cards` :

```
 curl -X GET
"<https://api.crustdata.com/screener/person/enrich?linkedin_profile_url=https://www.linkedin.com/in/sasikumarm00&enrich_realtime=true&fields=linkedin_profile_url,linkedin_flagship_url,name,location,email,title,last_updated,headline,summary,num_of_connections,skills,profile_picture_url,twitter_handle,languages,all_employers,past_employers,current_employers,education_background.degree_name,education_background.end_date,education_background.field_of_study,education_background.institute_linkedin_id,education_background.institute_linkedin_url,education_background.institute_logo_url,education_background.institute_name,education_background.start_date,all_employers_company_id,all_titles,all_schools,all_degrees,linkedin_open_to_cards,certifications,honors>" \\
-H "Authorization: Token auth_token" \\
-H "Content-Type: application/json"
```

○

4. Request without fields:

```
 curl -X GET
"<https://api.crustdata.com/screener/person/enrich?linkedin_profile_url=https://www.linkedin.com/in/sasikumarm00&enrich_realtime=true>" \\
-H "Authorization: Token auth_token" \\
-H "Content-Type: application/json"
```

○

5. Request with business email:

```
 curl -X GET
"<https://api.crustdata.com/screener/person/enrich?business_email=shubham.joshi@coindcx.com&enrich_realtime=true>" \\
```

-H "Authorization: Token auth_token" \\
-H "Content-Type: application/json"

- **Response:**
  - When LinkedIn profiles are present in Crustdata's database:
    - Response will include the enriched data for each profile. [JSON Hero]
  - When one or more LinkedIn profiles are not present in Crustdata's database:
    - An error message will be returned for each profile not found, along with instructions to query again after 60 minutes. https://jsonhero.io/j/kwdasun8HdqM
  - Response with all possible fields: https://jsonhero.io/j/zenKXWh36HsM
- **Notes**
  - If some profiles or emails are not found in the database and `enrich_realtime=False`, an empty response for those entries is returned, and they will be auto-enriched in the background. Query again after at least **60 minutes** to retrieve the data.
  - If `enrich_realtime=True` and the profile or email cannot be found even via real-time search, an error message is returned for those entries.
- **Key points:**

  Latency

  - **Database Search:** Less than **10 seconds** per profile.
  - **Real-Time Search:** May take longer due to fetching data from the web.
- **Limits**
  - **Profiles/Emails per Request:** Up to **25**.
  - **Exceeding Limits:** Requests exceeding this limit will be rejected with an error message.
- **Credits**
  - **Database Enrichment:**
    - **3 credits** per LinkedIn profile or email.
  - **Real-Time Enrichment (enrich_realtime=True):**
    - **5 credits** per LinkedIn profile or email.
- **Constraints**
  - **Valid Input:** Ensure all LinkedIn URLs and email addresses are correctly formatted.
    - Invalid inputs result in validation errors.
  - **Mutually Exclusive Parameters:** Do not include both linkedin_profile_url and business_email in the same request.
  - **Independent Processing:** Each profile or email is processed independently.
    - Found entries are returned immediately

        - Not found entries trigger the enrichment process (if enrich_realtime=False)

Search: LinkedIn People Search API (real-time)

Overview: Search for people profiles based on either a direct LinkedIn Sales Navigator search URL or a custom search criteria as a filter. This endpoint allows you to retrieve detailed information about individuals matching specific criteria.

Each request returns upto 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 Leads 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.

Making Requests

- **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 Leads search from the browser
    - `filters` (optional): JSON dictionary defining the search criteria as laid out by the [Crustdata filter schema](#).
    - `page` (optional): Page number for pagination (used only with `filters`)
    - `preview` (optional): Boolean field to get the preview of profiles. When using `preview` don't use `page`.
- **Examples**

Via LinkedIn Sales Navigator URL:

```
 curl --location '<https://api.crustdata.com/screener/person/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/people?query=(recentSearchParam%3A(id%3A394
0840412%2CdoLogHistory%3Atrue)%2Cfilters%3AList((type%3ACOMPANY_HEADCOUNT
%2Cvalues%3AList((id%3AC%2Ctext%3A11-50%2CselectionType%3AINCLUDED)%2C(id
%3AB%2Ctext%3A1-10%2CselectionType%3AINCLUDED)%2C(id%3AD%2Ctext%3A51-2
00%2CselectionType%3AINCLUDED)%2C(id%3AE%2Ctext%3A201-500%2CselectionType
%3AINCLUDED)%2C(id%3AF%2Ctext%3A501-1000%2CselectionType%3AINCLUDED)>))
%2C(type%3AINDUSTRY%2Cvalues%3AList((id%3A41%2Ctext%3ABanking%2Cselection
Type%3AINCLUDED)%2C(id%3A43%2Ctext%3AFinancial%20Services%2CselectionType
%3AINCLUDED)))%2C(type%3ACOMPANY_HEADQUARTERS%2Cvalues%3AList((id%3A
105912732%2Ctext%3ABelize%2CselectionType%3AINCLUDED)%2C(id%3A101739942%
2Ctext%3ACosta%20Rica%2CselectionType%3AINCLUDED)%2C(id%3A106522560%2Cte
xt%3AEl%20Salvador%2CselectionType%3AINCLUDED)%2C(id%3A100877388%2Ctext%
3AGuatemala%2CselectionType%3AINCLUDED)%2C(id%3A101937718%2Ctext%3AHond
uras%2CselectionType%3AINCLUDED)%2C(id%3A105517145%2Ctext%3ANicaragua%2C
selectionType%3AINCLUDED)%2C(id%3A100808673%2Ctext%3APanama%2CselectionTy
pe%3AINCLUDED)%2C(id%3A100270819%2Ctext%3AAntigua%20and%20Barbuda%2Cse
lectionType%3AINCLUDED)%2C(id%3A106662619%2Ctext%3AThe%20Bahamas%2Csele
ctionType%3AINCLUDED)%2C(id%3A102118611%2Ctext%3ABarbados%2CselectionType
%3AINCLUDED)%2C(id%3A106429766%2Ctext%3ACuba%2CselectionType%3AINCLUDE
D)%2C(id%3A105057336%2Ctext%3ADominican%20Republic%2CselectionType%3AINCL
UDED)%2C(id%3A100720695%2Ctext%3ADominica%2CselectionType%3AINCLUDED)%2
C(id%3A104579260%2Ctext%3AGrenada%2CselectionType%3AINCLUDED)%2C(id%3A1
00993490%2Ctext%3AHaiti%2CselectionType%3AINCLUDED)%2C(id%3A105126983%2Ct
ext%3AJamaica%2CselectionType%3AINCLUDED)%2C(id%3A102098694%2Ctext%3ASai
nt%20Kitts%20and%20Nevis%2CselectionType%3AINCLUDED)%2C(id%3A104022923%2
Ctext%3ASaint%20Lucia%2CselectionType%3AINCLUDED)%2C(id%3A104703990%2Ctex
t%3ASaint%20Vincent%20and%20the%20Grenadines%2CselectionType%3AINCLUDED)%
2C(id%3A106947126%2Ctext%3ATrinidad%20and%20Tobago%2CselectionType%3AINCL
UDED)%2C(id%3A107592510%2Ctext%3ABelize%20City%2C%20Belize%2C%20Belize%
2CselectionType%3AINCLUDED)))%2C(type%3ASENIORITY_LEVEL%2Cvalues%3AList((i
d%3A110%2Ctext%3AEntry%20Level%2CselectionType%3AEXCLUDED)%2C(id%3A100%
2Ctext%3AIn%20Training%2CselectionType%3AEXCLUDED)%2C(id%3A200%2Ctext%3A
Entry%20Level%20Manager%2CselectionType%3AEXCLUDED)%2C(id%3A130%2Ctext%
3AStrategic%2CselectionType%3AEXCLUDED)%2C(id%3A300%2Ctext%3AVice%20Presid
ent%2CselectionType%3AINCLUDED)%2C(id%3A220%2Ctext%3ADirector%2CselectionTy
pe%3AINCLUDED)%2C(id%3A320%2Ctext%3AOwner%20%2F%20Partner%2CselectionT
ype%3AINCLUDED)%2C(id%3A310%2Ctext%3ACXO%2CselectionType%3AINCLUDED)))
))&sessionId=UQyc2xY6ROisdd%2F%2B%2BsxmJA%3D%3D"
}'
```

○

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 working at `Google` or `Microsoft`, excluding those with the titles `Software Engineer` or `Data Scientist`, based in companies headquartered in `United States` or `Canada`, from the `Software Development` or `Hospitals and Health Care` industries, while excluding people located in `California, United States` or `New York, United States`

```
 curl --location '<https://api.crustdata.com/screener/person/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": "CURRENT_COMPANY",
            "type": "in",
            "value": ["Google", "Microsoft"]
        },
        {
            "filter_type": "CURRENT_TITLE",
            "type": "not in",
            "value": ["Software Engineer", "Data Scientist"]
        },
        {
            "filter_type": "COMPANY_HEADQUARTERS",
            "type": "in",
            "value": ["United States", "Canada"]
        },
        {
            "filter_type": "INDUSTRY",
            "type": "in",
            "value": ["Software Development", "Hospitals and Health Care"]
        },
        {
            "filter_type": "REGION",
            "type": "not in",
            "value": ["California, United States", "New York, United States"]
        }
    ],
    "page": 1
}'
```

- More Examples

1. People with specific first name from a specific company given company’s domain

```
 curl --location '<https://api.crustdata.com/screener/person/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": "FIRST_NAME",
      "type": "in",
      "value": ["steve"]
    },
    {
      "filter_type": "CURRENT_COMPANY",
      "type": "in",
      "value": ["buzzbold.com"]
    }
  ],
"page": 1
}'
```

○

2. People with specific first name from a specific company given company’s linkedin url

```
 curl --location '<https://api.crustdata.com/screener/person/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": "FIRST_NAME",
      "type": "in",
      "value": ["Ali"]
    },
    {
      "filter_type": "CURRENT_COMPANY",
      "type": "in",
      "value": ["<https://www.linkedin.com/company/serverobotics>"]
    }
  ],
"page": 1
}'
```

- ○

3. Preview list of people given filter criteria

```
 curl --location '<https://api.crustdata.com/screener/person/search>' \\
--header 'Content-Type: application/json' \\
--header 'Accept: application/json, text/plain, */*' \\
--header 'Authorization: Token $token' \\
--data '{"filters":[
    {
      "filter_type": "CURRENT_COMPANY",
      "type": "in",
      "value": ["serverobotics.com"]
    },
    {
      "filter_type": "REGION",
      "type": "in",
      "value": ["United States"]
    }
  ],
  "preview": true
}'
```

- ○

4. People that recently changed jobs and are currently working at a specific company

```
 curl --location '<https://api.crustdata.com/screener/person/search>' \\
--header 'Content-Type: application/json' \\
--header 'Accept: application/json, text/plain, */*' \\
--header 'Authorization: Token $token' \\
--data '{"filters":[
    {
      "filter_type": "CURRENT_COMPANY",
      "type": "in",
      "value": ["serverobotics.com"]
    },
    {
      "filter_type": "RECENTLY_CHANGED_JOBS"
    }
  ]
}'
```

- ○

- **Response**:
    - Default (without `preview=True`): https://jsonhero.io/j/t2CJ3nG7Xymv
    - With `preview=True` : https://jsonhero.io/j/yDSFQui0BKx8

- **Response with preview**

  https://jsonhero.io/j/V2VkhY4KrHSF

Key points:

- **Credits:** Each successful page request costs 25 credits. With `preview` , a successful request costs 5 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 `nth` page, `linkedin_sales_navigator_search_url` would become `https://www.linkedin.com/sales/search/people?page=n&query=....`

Example request with `page=2`

```
 curl --location '<https://api.crustdata.com/screener/person/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/people?page=2&query=(recentSearchParam%3A(i
d%3A3940840412%2CdoLogHistory%3Atrue)%2Cfilters%3AList((type%3ACOMPANY_HEA
DCOUNT%2Cvalues%3AList((id%3AC%2Ctext%3A11-50%2CselectionType%3AINCLUDE
D)%2C(id%3AB%2Ctext%3A1-10%2CselectionType%3AINCLUDED)%2C(id%3AD%2Ctext
%3A51-200%2CselectionType%3AINCLUDED)%2C(id%3AE%2Ctext%3A201-500%2Csele
ctionType%3AINCLUDED)%2C(id%3AF%2Ctext%3A501-1000%2CselectionType%3AINCL
UDED)>))%2C(type%3AINDUSTRY%2Cvalues%3AList((id%3A41%2Ctext%3ABanking%2
CselectionType%3AINCLUDED)%2C(id%3A43%2Ctext%3AFinancial%20Services%2Csele
ctionType%3AINCLUDED)))%2C(type%3ACOMPANY_HEADQUARTERS%2Cvalues%3ALi
st((id%3A105912732%2Ctext%3ABelize%2CselectionType%3AINCLUDED)%2C(id%3A101
739942%2Ctext%3ACosta%20Rica%2CselectionType%3AINCLUDED)%2C(id%3A1065225
60%2Ctext%3AEl%20Salvador%2CselectionType%3AINCLUDED)%2C(id%3A100877388%
2Ctext%3AGuatemala%2CselectionType%3AINCLUDED)%2C(id%3A101937718%2Ctext%
3AHonduras%2CselectionType%3AINCLUDED)%2C(id%3A105517145%2Ctext%3ANicara
gua%2CselectionType%3AINCLUDED)%2C(id%3A100808673%2Ctext%3APanama%2CseI
ectionType%3AINCLUDED)%2C(id%3A100270819%2Ctext%3AAntigua%20and%20Barbud
a%2CselectionType%3AINCLUDED)%2C(id%3A106662619%2Ctext%3AThe%20Bahamas
%2CselectionType%3AINCLUDED)%2C(id%3A102118611%2Ctext%3ABarbados%2Cselect
ionType%3AINCLUDED)%2C(id%3A106429766%2Ctext%3ACuba%2CselectionType%3AI
NCLUDED)%2C(id%3A105057336%2Ctext%3ADominican%20Republic%2CselectionType
%3AINCLUDED)%2C(id%3A100720695%2Ctext%3ADominica%2CselectionType%3AINCL
UDED)%2C(id%3A104579260%2Ctext%3AGrenada%2CselectionType%3AINCLUDED)%2
C(id%3A100993490%2Ctext%3AHaiti%2CselectionType%3AINCLUDED)%2C(id%3A10512
6983%2Ctext%3AJamaica%2CselectionType%3AINCLUDED)%2C(id%3A102098694%2Ct
ext%3ASaint%20Kitts%20and%20Nevis%2CselectionType%3AINCLUDED)%2C(id%3A104
022923%2Ctext%3ASaint%20Lucia%2CselectionType%3AINCLUDED)%2C(id%3A1047039
90%2Ctext%3ASaint%20Vincent%20and%20the%20Grenadines%2CselectionType%3AINC
LUDED)%2C(id%3A106947126%2Ctext%3ATrinidad%20and%20Tobago%2CselectionType
%3AINCLUDED)%2C(id%3A107592510%2Ctext%3ABelize%20City%2C%20Belize%2C%2
0Belize%2CselectionType%3AINCLUDED)))%2C(type%3ASENIORITY_LEVEL%2Cvalues
%3AList((id%3A110%2Ctext%3AEntry%20Level%2CselectionType%3AEXCLUDED)%2C(id
%3A100%2Ctext%3AIn%20Training%2CselectionType%3AEXCLUDED)%2C(id%3A200%2
Ctext%3AEntry%20Level%20Manager%2CselectionType%3AEXCLUDED)%2C(id%3A130
%2Ctext%3AStrategic%2CselectionType%3AEXCLUDED)%2C(id%3A300%2Ctext%3AVice
%20President%2CselectionType%3AINCLUDED)%2C(id%3A220%2Ctext%3ADirector%2C
selectionType%3AINCLUDED)%2C(id%3A320%2Ctext%3AOwner%20%2F%20Partner%2
CselectionType%3AINCLUDED)%2C(id%3A310%2Ctext%3ACXO%2CselectionType%3AIN
CLUDED)))))&sessionId=UQyc2xY6ROisdd%2F%2B%2BsxmJA%3D%3D"
}'
```

        -

    - When passing `filters` :
        - provide `page` as one of the keys in the payload itself

Example request with `page=1`

```
 curl --location '<https://api.crustdata.com/screener/person/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": "CURRENT_COMPANY",
         "type": "in",
         "value": ["Google", "Microsoft"]
      },
      {
         "filter_type": "CURRENT_TITLE",
         "type": "not in",
            "value": ["Software Engineer", "Data Scientist"]
        },
        {
            "filter_type": "COMPANY_HEADQUARTERS",
            "type": "in",
            "value": ["United States", "Canada"]
        },
        {
            "filter_type": "INDUSTRY",
            "type": "in",
            "value": ["Software Development", "Hospitals and Health Care"]
        },
        {
            "filter_type": "REGION",
            "type": "not in",
            "value": ["California, United States", "New York, United States"]
        }
    ],
    "page": 1
}'
```

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 person enrichment endpoint `screener/people/enrich` . But all the results will be added to Crustdata's database in 10 min of your query and the data for a specific person profile can be enriched via person enrichment endpoint

LinkedIn Posts by Person API (real-time)

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

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:

  - `person_linkedin_url` (required): LinkedIn profile url of the person. For example, any of these formats work [`https://linkedin.com/in/abhilash-chowdhary`](https://linkedin.com/in/abhilash-chowdhary) (flagship url) or [`https://linkedin.com/in/ACoAAAAsKtMBHQPJ9rgxpUs8M6pSxrAYCXIX8oY`](https://linkedin.com/in/ACoAAAAsKtMBHQPJ9rgxpUs8M6pSxrAYCXIX8oY) (fsd_profile url)
  - `fields` (optional): comma separated list of fields which you want to get in response.
    - all possible values:
      - total_reactions
      - total_comments
      - text
      - share_urn
      - share_url
      - reactors
      - reactions_by_type.PRAISE
      - reactions_by_type.LIKE
      - reactions_by_type.INTEREST
      - reactions_by_type.ENTERTAINMENT
      - reactions_by_type.EMPATHY
      - reactions_by_type.CURIOUS
      - reactions_by_type.APPRECIATION
      - reactions_by_type
      - num_shares
      - hyperlinks.person_linkedin_urls
      - hyperlinks.other_urls
      - hyperlinks.company_linkedin_urls
      - hyperlinks
      - date_posted
      - backend_urn
      - actor_name
      - year_founded
    - default: All fields except `reactors` :`total_reactions,total_comments,text,share_urn, share_url,reactions_by_type_PRAISE,reactions_by_type_LIKE,reactions_by_type_INTEREST,reactions_by_type_ENTERTAINMENT,reactions_by_type_EMPATHY,reactions_by_type_CURIOUS,reactions_by_type_APPRECIATION,reactions_by_type,num_shares,h`

                `yperlinks_person_linkedin_urls,hyperlinks_other_urls,hyperlinks_company_linkedin_urls,hyperlinks,date_posted,backend_urn,actor_name,year_founded`

        - `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)
            - All post types
                - `original`: only original posts are returned
                - `repost` : only reposted posts are returned
    - **Example Request:**

With default `fields` (without reactors)

```
 curl
'<https://api.crustdata.com/screener/linkedin_posts?person_linkedin_url=https://linkedin.com
/in/abhilash-chowdhary&page=1>' \\
  --header 'Accept: application/json, text/plain, */*' \\
  --header 'Accept-Language: en-US,en;q=0.9' \\
  --header 'Authorization: Token $auth_token'
```

-

With default `fields` and reactors

```
 curl
'<https://api.crustdata.com/screener/linkedin_posts?person_linkedin_url=https://linkedin.com
/in/abhilash-chowdhary&page=1&fields=reactors>' \\
  --header 'Accept: application/json, text/plain, */*' \\
  --header 'Accept-Language: en-US,en;q=0.9' \\
  --header 'Authorization: Token $auth_token'
```

-

With default `post_types`

```
 curl
'<https://api.crustdata.com/screener/linkedin_posts?person_linkedin_url=https://linkedin.com
/in/abhilash-chowdhary&page=1&post_types=post_types=repost%2C%20original>' \\
  --header 'Accept: application/json, text/plain, */*' \\
  --header 'Accept-Language: en-US,en;q=0.9' \\
  --header 'Authorization: Token $auth_token'
```

-

- **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: https://jsonhero.io/j/IGFH6zi5y9rP

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.\\n\\nHow do you gauge the buzz around these startups when most are pre-product?\\n\\nWe’ve defined web traffic as the metric to go by.\\n\\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\\n🤔 Interested in reading more about the YC S24 batch?\\n\\nRead our full breakdown from the link in the comments 👇",
      "actor_name": "Crustdata",
      "hyperlinks": {
         "company_linkedin_urls": [],
         "person_linkedin_urls": [
            "<https://www.linkedin.com/in/ACoAAAKoldoBqSsiXY_DHsXdSk1slibabeTvDDY>"
         ],
         "other_urls": []
      },
      "date_posted": "2024-09-03",
      "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": [
    {
     "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=vHg233746zA00m3q2vHKSFcthL3YKiagTtVEZt1qqJI>",
     "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.
- `date_posted` (string): Date when the post was published, in "YYYY-MM-DD" format.
- `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
- `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).
- linkedin_headcount_and_glassdoor_ceo_approval_and_g2
- `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.
- `linkedin_profile_urn` (string): Unique identifier for the reactor's LinkedIn profile.
- `default_position_title` (string): Primary job title.
- `default_position_company_linkedin_id` (string): LinkedIn ID of the reactor's primary company.
- `default_position_is_decision_maker` (boolean): Indicates if the reactor is in a decision-making role.
- `flagship_profile_url` (string): Another form of the reactor's LinkedIn profile URL.
- `profile_picture_url` (string): URL to a larger version of the profile picture (400x400 size).
- `headline` (string): Professional headline from the reactor's LinkedIn profile.
- `summary` (string or null): Brief professional summary, if available.
- `num_of_connections` (integer): Number of LinkedIn connections the reactor has.
- `related_colleague_company_id` (integer): LinkedIn ID of a related company, possibly current employer.
- `skills` (array of strings): List of professional skills listed on the reactor's profile.
- `employer` (array of objects): Employment history, each containing:
  - `title` (string): Job title.
  - `company_name` (string): Name of the employer.
  - `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`.

- **Key Points**
    - **Credits:**
        - Without reactors (default): Each successful page request costs 5 credits
        - With reactors: Each successful page request costs 25 credits
    - **Pagination:**
        - Increment the value of `page` query param to fetch the next set of posts.
        - Most recent posts will be in first page and then so on.
        - Currently, you can only fetch only upto 20 pages of latest posts. In case you want to fetch more, contact Crustdata team at info@crustdata.com .
    - External urls or Company/Person LinkedIn urls mentioned in text:
        - `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 depending on number of reactions for all the posts in the page

LinkedIn Posts Keyword Search (real-time)

API Usage Endpoints

Get remaining credits

Request

A plain GET request without any query params.

Required: authentication token `auth_token` for user identification.

```
 curl --location '<https://api.crustdata.com/user/credits>' \\
--header 'Accept: application/json, text/plain, */*' \\
--header 'Accept-Language: en-US,en;q=0.9' \\
--header 'Authorization: Token $auth_token' \\
--header 'Content-Type: application/json'
```

-

Response

Returns the remaining credits for the current billing period

```
 {
   "credits": 1000000
}
```

-