API OVERVIEW

API REFERENCE

Lever API Overview

Introduction

The root URL for the Lever API web service is https://api.lever.co/v1.

All data is sent and received as JSON over HTTPS, using the Unicode UTF-8 text encoding.

Rest assured, we won’t rename or remove fields without a version bump. But we may decide to add additional fields to the API in the future.

Use Cases

The Lever API lets you build a variety of applications and scripts to create candidates, integrate Lever with other business platforms (HRIS, HRMS, etc.), and show Lever data outside of the app.

Hoping to build an integration with your jobs site? You're probably looking for the Postings API.

Using our candidate creation and update endpoints, you can build:

- A custom jobs page
- A custom referral / agency portal
- A university-specific jobs portal
- Sync to HRIS/HRMS to update candidate information in Lever

The Lever API also allows you to integrate with third-party tools to:

- Add a candidate into Lever with custom sources, tags, stage, and meta data; with or without a job posting
- Update a candidate status in Lever based on events in your third-party app
- Add sources and tags to a candidate profile in Lever

Here are some examples of the things you can do with the Lever API:

- Export candidate data to onboarding and background check software
- Export richer data downloads than those that are available through Lever’s UI
- Pull data out of Lever for custom reports
- An HRIS/HRMS integration that pulls candidate data about all hired candidates
- A dashboard page that shows a visual representation of candidate data

Authentication

You authenticate to the Lever API by providing one of your API keys in the request. You can manage your API keys from the Integrations tab in Settings. You can have multiple

API keys active at one time. Your API keys carry many privileges, so keep them secret!

For each request, provide your API key as the basic auth username (leave the password blank). You must authenticate for all requests and all API requests must be made over HTTPS. That is, unencrypted HTTP is not supported.

Rate Limits

We currently rate limit by API key using an implementation of token bucket. At this moment in time, we don't limit at the individual endpoint level. By default, we allow a steady state number of 10 requests/second per API key. If available, we allow bursts of requests up to 20 per second. These defaults are not guaranteed, as they may vary with server load and we may change them in the future. We recommend retrying using exponential backoff.

Rate limits are important to prevent abuse and keep the service available to everyone.

Responses and Errors

Sometimes requests to the API are not successful. Failures can occur for a wide range of reasons. In all cases, the API should return an HTTP Status Code that will indicate the nature of the failure, with a response body in JSON format containing additional information.

200 OK	If data was requested, it will be available in the `data` field at the top level of the response body.
201 Created	201 Created - Resource was successfully created and information is available in the data field at the top level of the response body.
400 Invalid Request	This usually occurs because of a missing or malformed parameter. Check the documentation and the syntax of your request and try again.
401 Unauthorized	A valid API key was not provided with the request. You must authenticate for all requests. Learn more about how authentication works here.
403 Forbidden	Your Lever account settings don't authorize you to perform the requested operation. Talk to a Super Admin on your Lever account to update your API settings.
404 Not Found	Either the request method and path supplied do not specify a known action in the API, or the object specified by the request does not exist.
429 Too Many Requests	Lever imposes a limit of the number of requests a client can make in a short time. Read more about our rate limiting here.
500 Server Error	Oops! There was an error on Lever's end.

Example error response

```
{
  "code": "ResourceNotFound",
  "message": "stages 578c0418-96be-4a78-80ef-4da5701de3c8 was not four
}
```

Pagination

All resources with a list endpoint (candidates, users, postings) have pagination support. By default, the limit is set to 100 results.

Pagination parameters:

limit

`50`

optional

A limit on the number of objects to be returned. The limit can range between 1 and 100 items. If no limit is specified, the default for that endpoint is used.

offset

`0.1414895548650.b9efe311-ee1a-440e-bb17-c5c0aed9eb3a`

optional

An offset token specifying the next page of results to return. A paginated list response will include a `next` attribute that includes an offset token, which can be used as an input parameter to the next request. If an offset is not passed in, the API will fetch the first page of results. You can only pass in an offset that was returned to you via a previously paginated request.

Lists

In addition to supporting pagination parameters, all list endpoints share a common format.

data

Array

An array of request resource objects

next

`0.1414895548650.b9efe311-ee1a-440e-bb17-c5c0aed9eb3a`

String

Offset token for next page

hasNext

`true`

boolean

Whether or not there are more elements available. If false, this is the end of the list.

```
{
  "data": {...},
  "next" : "0.1414895548650.b9efe311-ee1a-440e-bb17-c5c0aed9eb3a",
  "hasNext": true
}
```

Customizing the Response

You can customize the response from an endpoint in two ways: you can adjust the attributes included in a response with the `include` parameter and you can specify which attributes are included as full objects with the `expand` parameter.

include `/postings?include=content`

optional

All list endpoints support the use of an include parameter to customize the inclusion of fields. By default all fields are included in responses. You can customize the fields included in the response in both lists and singular requests by specifying exactly what fields are desired with `include`. If the `include` parameter is specified, no other fields other than those set in the request will be returned. To include multiple optional fields use `/postings?include=content&include=followers`

expand `/postings?expand=followers`

optional

Many objects contain the ID of another object in their response properties. Those objects can be expanded inline with the expand request parameter. Objects that can be expanded are noted in this documentation. This parameter is available on all API requests and applies to the response of that request only. For example, requesting the users who are followers of a specific candidate will expand the followers property into an array of full user objects. Note: If the `expand` option is also used, it will take precedence over the include option and anything that is expanded will automatically be included.

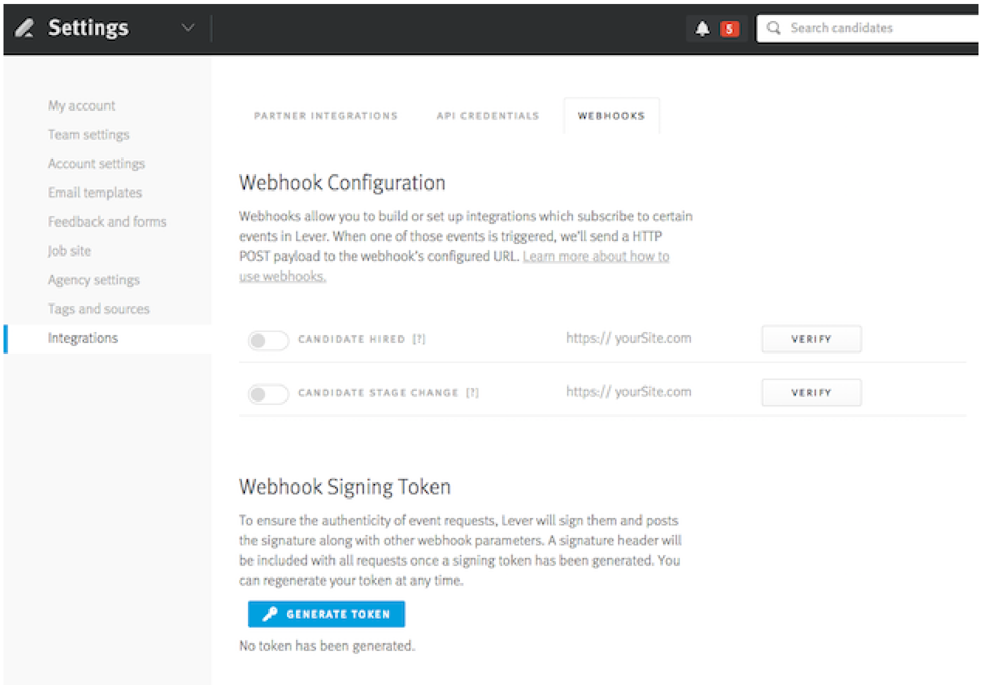
Webhooks

Webhooks allow you to build or set up integrations which subscribe to certain events in Lever. When one of those events is triggered, we'll send a HTTP POST payload to the webhook's configured URL.

You can specify a custom webhook URL for each event type in Lever. We also offer one-click integrations with our partners. To see what partner integrations are available, visit your account integration settings.

How to set up webhooks

To configure webhooks for your account, you must be a Super Admin. To get started, visit the webhooks tab in your account integration settings.



Webhook Configuration

Webhooks allow you to build or set up integrations which subscribe to certain events in Lever. When one of those events is triggered, we'll send a HTTP POST payload to the webhook's configured URL. Learn more about how to use webhooks.

Webhook Signing Token

To ensure the authenticity of event requests, Lever will sign them and posts the signature along with other webhook parameters. A signature header will be included with all requests once a signing token has been generated. You can regenerate your token at any time.

No token has been generated.

Here you can specify exactly what events you would like to enable webhooks for. You can also test your webhook URL from this page. Lever will send a sample request and will indicate whether the request succeeds (2xx) or fails. Our test POST request looks like this:

```
POST /yourWebhookUrl HTTP/1.1

User-Agent: "Lever-Webhooks"
Content-Type: "application/json"
Content-Length: 202

{
 "triggeredAt": 1431458021951,
 "event": "candidateHired",
 "data": {},
 "signature": "d3e73080552bcbf65f66c90b6d4ac0db664c6ad762efb1899653808
 "token": "457fd5a0759c87d25460b096dbc388d1d21058849f08ce2f"
}
```

If your test request fails, you can view more details about the request and response in the webhooks delivery history.

If your test request happens to fail because you have not set up your service yet or it is not yet running, do not worry. We will still send webhooks to your configured URL even if the test request fails. But please do try and enter a valid URL.

Securing webhooks

We only support HTTPS endpoints

Since we are sending confidential candidate information outside of Lever, we need to ensure all traffic is encrypted. For this reason, we only support HTTPS enabled endpoints. However, we do not verify the SSL certificate against a Certificate Authority and, as such, the use of self-signed SSL certificates is supported. For local development, we suggest using a reverse proxy that supports HTTPS termination such as ngrok.

Request signing

To ensure the authenticity of webhook requests, Lever signs them and includes the signature within the body of the POST request. A signature body parameter will be included with all requests if a signing token has been generated in the integrations settings admin page. You can regenerate your signing token at any time.

Lever uses the follow request parameters to generate the signature:

triggeredAt	`1431458021951` When webhook event was triggered
token	`457fd5a0759c87d25460b096dbc388d1d21058849f08ce2f` Randomly generated string with length of 50 characters

To verify the webhook is originating from Lever you need to:

- Concatenate token and triggeredAt values
- Encode the resulting string with the HMAC algorithm (using your signature token as a key and SHA256 digest mode)
- Compare the resulting hexdigest to the signature

Optionally, you can cache the token value locally and not honor any subsequent request with the same token. This will prevent replay attacks. You can also check if the timestamp is not too far from the current time.

Below is a Javascript code sample used to verify the signature:

```
function validateWebhook(body) {
  var plainText = body.token + body.triggeredAt
  var hash = crypto.createHmac('sha256', SIGNATURE_TOKEN)
  hash.update(plainText)
  hash = hash.digext('hex')
  return body.signature === hash
}
```

Receiving webhooks

With webhooks, your server is the server receiving the request. Configuring your server to receive a new webhook is no different from creating any other route. You can configure any URL you would like to receive events. Webhooks events are sent as POST requests with the data sent as a JSON object on the request body.

Each event has a similar JSON schema, but a unique `data` object that is determined by its event type.

id	Event UID Unique ID for this event
event	`candidateHired` String event type

triggeredAt
`1431458021951`
Timestamp [?]
When webhook event was triggered

token
`457fd5a0759c87d25460b096dbc388d1d21058849f08ce2f`
String
Randomly generated string with length 50

signature
`457fd5a0759c87d25460b096dbc388d1d21058849f08ce2f`
String
Randomly generated string with length 50

data
Object
Data payload custom for each event. See Event payloads for details.

Event payloads

The following is a list of the different events you can enable from the integration settings page. It also provides additional detail on what properties will be provided within each of the event's `data` object.

Candidate hired
`event: candidateHired`
Triggers when a candidate is marked as hired in Lever.

+ Show child fields

Candidate stage change
`event: candidateStageChange`
Triggers when a candidate moves from one stage to another in Lever.

+ Show child fields

Candidate archive state change
`event: candidateArchiveChange`
Triggers when a candidate is archived or unarchived in Lever.

+ Show child fields

Example delivery

```
POST /yourWebhookUrl HTTP/1.1

User-Agent: "Lever-Webhooks"
Content-Type: "application/json"
Content-Length: 373

{
  "triggeredAt": 1431454330072,
  "event": "candidateStageChange",
  "data": {
    "candidateId": "e80dbe7b-ea7d-4917-8342-bc183e5eb2dc",
    "previousStageId": "e2d047a9-f313-4220-8412-7f8e7838564a",
    "currentStageId": "0a570865-0ca0-4562-b681-653126d98154"
  },
  "signature": "d150a0eea6291508027223135f37ba4dceb784b52bfa2f6d985d3f
  "token": "7b530f499295b30cda86ed7d288cd1a808e635a77abc6bec"
}
```

Responding to a webhook

To acknowledge you received the webhook without any problem, your server should return a 2xx HTTP status code. Any other information you return in the request headers or request body will be ignored. Any response code outside the 2xx range, including 3xx codes, will indicate to Lever that you did not receive the webhook. When a webhook is not received for whatever reason, Lever will continue trying to send the webhook five times, waiting longer and longer in between tries.

Webhooks delivery history

We show the most recent webhooks from (up to 1,000 requests) from the past two weeks on the webhook configuration page. If you click on one of the deliveries, the detail view for that delivery will open. Here you will be able to see request and response associated with that delivery. Furthermore, if the webhook request failed, you can view additional information about the error here.

Webhook requests can also be rerun manually by clicking on the 'Rerun' button from within the detail view.

Lever API Reference

A reference for the types of data you can access through the Lever API.

Applications

When a candidate is applied to a job posting, an application is created.

Lever is candidate-centric, meaning that candidates can exist in the system without being applied to a specific job posting. However, almost all candidates are applied to job postings, and thus almost all candidates have one or more applications.

There are three different ways that applications can be created in Lever

- Through a posting: Application is created when a candidate applies to a published job posting through your company's careers page.
- By a user: A team member at your company manually added a job posting to a specific candidate in Lever.
- As a referral: A team member at your company referred the candidate into Lever for a specific job posting.

Candidates can be applied to multiple job postings, meaning that candidates can have multiple applications. Yet since Lever is candidate-centric, each of these applications will be on the same candidate profile.

Attributes

id	`6ffe4153-60ba-4e30-afae-bd9a9775879c` String Application UID
candidateId	`c0950a08-e88e-4435-aada-ce8119a819ad` String

Candidate profile associated with an application.

createdAt
`1418353241665`
Timestamp
Datetime when application was created in Lever

type
`posting`
String
An application can be of type `referral`, `user`, or `posting`. Applications of type `referral` are created when a user refers a candidate for a job posting. Applications have type `user` when they are applied manually to a posting in Lever. Applications have type `posting` when a candidate applies to a job posting through your company's jobs page.

posting
`dc8816a3-d7b9-4068-ad23-821a93a8f710`
Posting UID
Job posting to apply to candidate
Expandable If expanded, contains a posting object

postingOwner
User UID
The owner of the job posting at the time when the candidate applies to that job.
Expandable If expanded, contains a user object

postingHiringManager
`eec54734-e912-4959-bf60-d9202f5ca916`
User UID
The hiring manager of the job posting at the time when the candidate applies to that job.
Expandable If expanded, contains a user object

user
User UID
If application is a referral, this is the user who made the referral.
Expandable If expanded, contains a user object

name
`Teresa Crile`
String
Name of candidate who applied

email
`teresa@gmail.com`
String
Candidate email

phone
`{"type":null,"value":"(123) 456-7888"}`
Object
Candidate phone number

company
String
Candidate's current company or organization

links
`null`
Array
List of candidate links (e.g. personal website, LinkedIn profile, etc.)

comments
String
Any additional comments from candidate included in job application

resume

WARNING: This field is deprecated. Use the resumes endpoint.

customQuestions

Array of forms

An array of customized forms. If application is type `referral`, the custom questions will include a referral form. If application is type `posting`, the custom questions include customized posting forms.

archived

Object

Application archived status.

+ Show child fields

requisitionForHire

Object

If the application was archived as hired against a requisition, this is the data related to the requisition.

+ Show child fields

Retrieve a single application

This method returns the full application record for a single application.

```
GET /candidates/:candidate/applications/:application
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": {
    "id": "6ffe4153-60ba-4e30-afae-bd9a9775879c",
    "candidateId": "c0950a08-e88e-4435-aada-ce8119a819ad",
    "createdAt": 1418353241665,
    "type": "posting",
    "posting": "dc8816a3-d7b9-4068-ad23-821a93a8f710",
    "user": null,
    "name": "Teresa Crile",
    "email": "teresa@gmail.com",
    "phone": {
      "type": null,
      "value": "(123) 456-7888"
    },
    "requisitionForHire": {
      "id": "498a7320-9715-452b-b688-0cd7bc908d88",
      "requisitionCode": "OP-9",
      "hiringManagerOnHire": "60c8c233-563e-4c27-8f42-7f1e078e1770"
    },
    "ownerId": "b830157a-1fe9-4dd8-890c-86401f9f5de5",
    "hiringManager": "eec54734-e912-4959-bf60-d9202f5ca916",
    "company": null,
    "links": null,
    "comments": "Dear Mr Black,\r\n\r\nPlease find enclosed my CV in a
    "resume": null,
    "customQuestions": [
      {
        "type": "posting",
        "text": "Software Engineer",
        "description": "",
        "fields": [
          {
            "type": "multiple-select",
            "text": "What languages/frameworks are you comfortable wit
            "description": "",
            "required": false,
            "options": [
              {
                "text": "javascript"
              },
              {
                "text": "node"
              },
              {
                "text": "derby"
              }
            ],
            "value": [
              "javascript",
              "node"
            ]
          },
          {
            "type": "file-upload",
            "text": "Please attach the programming assignment",
            "description": "",
            "required": false,
            "value": {
              "id": "38d5b3fa-3aa9-4b9d-917b-e48b02685617",
              "ext": ".txt",
              "name": "derby-hw.txt",
              "size": 46,
              "uploadedAt": 1423509968652
            }
          },
          {
            "type": "text",
            "text": "What role most interests you?",
            "description": "",
            "required": false,
            "value": " I'm interested in primarily a frontend role"
          }
        ],
        "baseTemplateId": "076c3cc4-b28b-452d-b8c0-4107555aa28d",
        "user": null,
        "stage": null,
        "createdAt": 1423176554644,
        "completedAt": null
      }
    ]
  }
}
```

List all applications

Lists all applications for a candidate.

```
GET /candidates/:candidate/applications
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": [
    {
      "id": "15c55fc3-e65a-403a-b882-2ee917a7149b",
      "candidateId": "c0950a08-e88e-4435-aada-ce8119a819ad",
      "createdAt": 1417586757232,
      "type": "user",
      "posting": "cf02e6ad-ddf8-4b15-b3bf-0aaa27d4940e",
      "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
      "name": null,
      "email": null,
      "phone": null,
      "company": null,
      "links": null,
      "comments": null,
      "resume": null,
      "customQuestions": null
    },
    {
      "id": "6ffe4153-60ba-4e30-afae-bd9a9775879c",
      "candidateId": "c0950a08-e88e-4435-aada-ce8119a819ad",
      "createdAt": 1418353241665,
      "type": "posting",
      "posting": "dc8816a3-d7b9-4068-ad23-821a93a8f710",
      "user": null,
      "name": "Teresa Crile",
      "email": "teresa@gmail.com",
      "phone": {
        "type": null,
        "value": "(123) 456-7888"
      },
      "requisitionForHire": {
        "id": "498a7320-9715-452b-b688-0cd7bc908d88",
        "requisitionCode": "OP-9",
        "hiringManagerOnHire": "60c8c233-563e-4c27-8f42-7f1e078e1770"
      },
      "ownerId": "b830157a-1fe9-4dd8-890c-86401f9f5de5",
      "hiringManager": "eec54734-e912-4959-bf60-d9202f5ca916",
      "company": null,
      "links": null,
      "comments": "Dear Mr Black,\r\n\r\nPlease find enclosed my CV ir
      "resume": null,
      "customQuestions": [
        {
          "type": "posting",
          "text": "Software Engineer",
          "description": "",
          "fields": [
            {
              "type": "multiple-select",
              "text": "What languages/frameworks are you comfortable w
              "description": "",
              "required": false,
              "options": [
                {
                  "text": "javascript"
                },
                {
                  "text": "node"
                },
                {
                  "text": "derby"
                }
              ],
              "value": [
                "javascript",
                "node"
              ]
            },
            {
              "type": "file-upload",
              "text": "Please attach the programming assignment",
              "description": "",
              "required": false,
              "value": {
                "id": "38d5b3fa-3aa9-4b9d-917b-e48b02685617",
                "ext": ".txt",
                "name": "derby-hw.txt",
                "size": 46,
                "uploadedAt": 1423509968652
              }
            },
            {
              "type": "text",
              "text": "What role most interests you?",
              "description": "",
              "required": false,
              "value": " I'm interested in primarily a frontend role"
            }
          ],
          "baseTemplateId": "076c3cc4-b28b-452d-b8c0-4107555aa28d",
          "user": null,
          "stage": null,
          "createdAt": 1423176554644,
          "completedAt": null
        }
      ]
    },
    {
      "id": "c9b3ce10-a677-4d21-a6a9-502b3d783928",
      "candidateId": "c0950a08-e88e-4435-aada-ce8119a819ad",
      "createdAt": 1418354167890,
      "type": "referral",
      "posting": "dc8816a3-d7b9-4068-ad23-821a93a8f710",
      "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
      "name": "Teresa Crile",
      "email": "teresa@gmail.com",
      "phone": null,
      "company": null,
      "links": null,
      "comments": null,
      "resume": null,
      "customQuestions": [
        {
          "accountId": "ba1bfaa7-f32d-4538-9c2c-e85b2b303a17",
          "createdAt": 1406930579778,
          "text": "Referral",
          "description": "",
          "type": "referral",
          "fields": [
            {
              "type": "text",
              "required": true,
              "text": "Name of referrer",
              "description": "",
              "options": [
                {
                  "text": ""
                }
              ],
              "isSummary": true,
              "summaryText": "Referred by",
              "value": "Josh Ackerman"
            },
            {
              "type": "dropdown",
              "required": true,
              "text": "Relationship",
              "description": "",
              "options": [
                {
                  "text": "Former colleague"
                },
                {
                  "text": "Friend"
                },
                {
                  "text": "Reputation"
                },
                {
                  "text": "Other"
                },
                {
                  "text": "Don't know this person"
                }
              ],
              "prompt": "Select one",
              "isSummary": true,
              "value": "Reputation"
            },
            {
              "type": "textarea",
              "required": true,
              "text": "Notes / Comments",
              "description": "",
              "options": [
                {
                  "text": ""
                }
              ],
              "value": "Heard about her work online."
            }
          ],
          "baseTemplateId": "1655e26a-847e-4eb6-86b2-4658a022b536",
          "referrerId": "d2b18d51-d4a0-4976-a614-9e481577892a",
          "userId": "d2b18d51-d4a0-4976-a614-9e481577892a"
        }
      ]
    }
  ]
}
```

Archive Reasons

Archive reasons provide granularity behind to candidates who have exited your active hiring pipeline. Candidates exit your active pipeline due to either being hired at your company or due to being rejected for a specific reason. These dispositions allows you to track each and every candidate who is no longer active within your pipeline. Check out this article for further information about Archive reasons.

Attributes

id `141b14d3-1d1a-4564-9b87-d0b3d12dd948`
String
Archive reason UID

text `Underqualified`
String
The name of the archive reason as shown in the Lever interface.

Retrieve a single archive reason

```
GET /archive_reasons/:archive_reason
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/archive_reasons/141b14d3-1d1a
```

```
{
  "data": {
    "id": "141b14d3-1d1a-4564-9b87-d0b3d12dd948",
    "text": "Underqualified"
  }
}
```

List all archive reasons

Lists all archive reasons in your Lever account.

```
GET /archive_reasons
```

Parameters

hired `true`
Optional
Filter for hired archive reasons

Examples

```
curl -u API_KEY: https://api.lever.co/v1/archive_reasons
```

```
{
  "data": [
    {
      "id": "141b14d3-1d1a-4564-9b87-d0b3d12dd948",
      "text": "Underqualified"
    },
    {
      "id": "0deb0487-b664-476e-9653-dfe8881ef8ff",
      "text": "Culture Fit"
    },
    {
      "id": "e825cceb-00fb-4e7d-b5bb-9f69ab8a08c4",
      "text": "Timing"
    },
    {
      "id": "f5e5364c-8653-4160-aa27-42f5d5db884f",
      "text": "Withdrew"
    },
    {
      "id": "d7cf6a64-3c69-4b3f-b770-9f10878b224c",
      "text": "Offer declined"
    },
    {
      "id": "b53daa04-8fc3-414b-accc-d26e7f6f669e",
      "text": "Hired"
    },
    {
      "id": "28774389-dbe4-4680-acf6-9cbbfed23849",
      "text": "Position filled"
    }
  ]
}
```

```
curl -u API_KEY: https://api.lever.co/v1/archive_reasons?type=hired
```

```
{
  "data": {
    "id": "b53daa04-8fc3-414b-accc-d26e7f6f669e",
    "text": "Hired"
  }
}
```

Audit Events

> Please note this endpoint (`/audit_events`) is an add-on for Enterprise customers. If you're interested in accessing the Audit Event endpoints for your account, please reach out to your customer success manager or drop our team a note.

Lever tracks certain activity by recording audit events. This endpoint exposes those events to help security teams monitor sensitive activity within Lever and investigate past activity when there may have been a violation.

Broadly, Lever's audit events track the following (see Tracked actions for the full list):

- User provisioning: *Why does a user have the access they have?*
- User authentication: *Who is trying to access the system, and when?*

- Data export: *How is our data leaving the system?*

Attributes

id

`5eca840d-3f89-48b7-b718-e3f655954959`

String

Audit Event UID

createdAt

`1412096173299`

Timestamp [?]

Time when the event was recorded (milliseconds since Unix epoch)

type

`user:added`

String

The tracked action. See Tracked actions for the list of values.

user

Object

User who performed the action.

+ Show child fields

target

Object

The resource the action was performed on.

+ Show child fields

meta

Object

Additional data about the event. Each `type` records different data in `meta` with different fields. See Event details for more information.

List all audit events

Lists all audit events in your Lever account, sorted in descending chronological order (newest first).

```
GET /audit_events
```

Parameters

type

`user:added`

String

Optional

Filter returned events to a single event type.

user_id

`13279cd8-f406-432d-8e8a-bb7e7afa85`

String

Optional

Filter returned events by the user who performed the action. To find events with no associated user (e.g. failed logins), set the value to `"null"`.

target_type

`"user"`

String

Optional

Filter returned events to actions on a particular kind of resource. This value should match the first part of a `type`.

target_id

`13279cd8-f406-432d-8e8a-bb7e7afa8`

String

Optional

Filter returned events to actions to a particular resource. The request must also specify a `target_type` to use this parameter.

created_at_start

`1412012000000`

Timestamp [?]

Optional

Filter returned events to the ones which occurred at or after a particular time.

created_at_end

`1412012800000`

Timestamp [?]

Optional

Filter returned events to the ones which occurred at or before a particular time.

Example

```
curl -u API_KEY: https://api.lever.co/v1/audit_events
```

```
{
  "data": [
    {
      "id": "3dd389df-75e4-4764-a69e-7d8c8d9c2d5a",
      "createdAt": 1454699926185,
      "type": "user.authentication:failed",
      "user": {
        "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
        "email": "nelly@brickly.com",
        "name": "Nelly Newton",
        "role": "interviewer"
      },
      "target": {
        "id": "13279cd8-f406-432d-8e8a-bb7e7afa8",
        "type": "user",
        "label": "Nelly Newton"
      },
      "meta": {
        "authentication": {
          "method": "direct",
          "error": {
            "message": "Looks like you don't have permission to access
            "type": "user-deactivated"
          }
        },
        "user": {
          "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
          "email": "nelly@brickly.com",
          "name": "Nelly Newton",
          "role": "admin"
        }
      }
    },
    {
      "id": "5eca840d-3f89-48b7-b718-e3f655954959",
      "createdAt": 1412096173299,
      "type": "user:added",
      "user": {
        "id": "13279cd8-f406-432d-8e8a-bb7e7afa85",
        "email": "nic@brickly.com",
        "name": "Nicolas Cage",
        "role": "super admin"
      },
      "target": {
        "id": "13279cd8-f406-432d-8e8a-bb7e7afa8",
        "type": "user",
        "label": "Nelly Newton"
      },
      "meta": {
        "user": {
          "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
          "email": "nelly@brickly.com",
          "name": "Nelly Newton",
          "role": "interviewer"
        }
      }
    }
  ]
}
```

Tracked actions

This is the full list of actions tracked by audit events.

TYPE	DESCRIPTION	
`account:added`	Account was created.	details
`account:deactivated`	Account was deactivated.	details
`account:reactivated`	Account was reactivated.	details
`account.authentication:changed`	Account authentication method changed.	details
`account.plan:changed`	Account plan was changed.	details
`account.suites:changed`	Account suite configuration was changed.	details
`export:downloaded`	Account data was downloaded to a file.	details
`key:added`	An API key was added.	details
`key:removed`	An API key was deleted.	details
`user:added`	A new user was added to the account.	details
`user:deactivated`	A user was deactivated.	details
`user:merged`	Two users were merged.	details
`user:reactivated`	A user was reactivated.	details
`user:removed`	A user was deleted.	details

TYPE	DESCRIPTION	
`user.authentication:failed`	A sign-in attempt failed.	details
`user.authentication:succeeded`	A user signed in.	details
`user.email:changed`	A user's email address was changed.	details
`user.linkedProfile:changed`	A user's linked profile was changed.	details
`user.password:changed`	A user changed their password.	details
`user.password:forgot`	A user requested a password recovery.	details
`user.permission:changed`	A change in a permission for a user.	details
`user.role:changed`	A user's access was changed.	details

Event details

Each type of event has a slightly different structure in `meta`.

account:added

Account was created by a Lever employee.

```
"meta": {
  "account": {
    "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
    "name": "Brickly"
  }
}
```

account:deactivated

Account was deactivated by a Lever employee due to a support request.

```
"meta": {
  "account": {
    "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
    "name": "Brickly"
  }
}
```

account:reactivated

Account was reactivated by a Lever employee due to a support request.

```
"meta": {
  "account": {
    "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
    "name": "Brickly"
  }
}
```

account.authentication:changed

Account authentication configuration was changed by Lever employee due to a support request.

```
  "meta": {
    "account": {
      "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
      "name": "Brickly"
    },
    "from": {
      "googleApps": true,
      "direct": false,
      "saml": false
    },
    "to": {
      "googleApps": true,
      "direct": false,
      "saml": true
    }
  }
```

account.plan:changed

Account plan was changed by Lever employee due to a support request.

```
  "meta": {
    "account": {
      "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
      "name": "Brickly"
    },
    "from": "Enterprise",
    "to": "Professional"
  }
```

account.suites:changed

Account suite configuration was changed by Lever employee due to a support request.

An account's suite configuration defines the authorization provider, calendar provider, and email provider for each of their domains (such as `google` for G Suite). Most accounts will use same provider across features and use the default configuration for that provider, but Lever also allows a `custom` suite provider option where each feature (authorization, email, and calendar) are configured separately.

```
  "meta": {
    "account": {
      "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
      "name": "Brickly"
    },
    "from": [
      {
        "domain": "brick.ly",
        "provider": "custom"
      }
    ],
    "to": [
      {
        "domain": "brickly.com",
        "provider": "google"
      }
    ]
  }
```

export:downloaded

A company admin downloaded a report’s data (with “Export to spreadsheet”) or other account data (like an interview feedback export) as a CSV file. The `rows` property is the number of non-header rows in the downloaded report.

```
  "meta": {
    "export": {
      "id": "1dccc190-9e72-4a4e-9eea-84bd2a7642ee",
      "name": "1dccc190-9e72-4a4e-9eea-84bd2a7642ee.all-postings.csv",
      "rows": 45
    }
  }
```

key:added

A user added a new API key for the Postings or Data API (`"postings-api"` or `"data-api"`). If the key was added for a partner from the account's Integration settings, then `"partner"` will be set to `true`.

```
  "meta": {
    "key": {
      "id": "12e4dd1f-5a13-496d-84db-156da2b3d4c0",
      "name": "Sketchy key",
      "partner": false,
      "tokenLastFour": "mLoi",
      "service": "data-api"
    }
  }
```

key:removed

A user removed an API key.

```
  "meta": {
    "key": {
      "id": "33d5b433-a9de-410d-9bac-1f542fb1f535",
      "name": "Namely",
      "partner": true,
      "tokenLastFour": "K1o2",
      "service": "data-api"
    }
  }
```

user:added

A new user was added to the account. This can happen in four ways:

- A new user is invited by a company admin.
- A new user is scheduled to conduct an interview by a company hiring manager or admin adds their email to the interview panel (The new user is created as an `"interviewer"` — the lowest level of access)

- A user signs in for the first time. The new user is created as an `"interviewer"` — the lowest level of access. (Only for accounts using Google Auth)
- A new user is added by a Lever employee due to a support request.

```
"meta": {
  "user": {
    "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
    "email": "nelly@brickly.com",
    "name": "Nelly Newton",
    "role": "interviewer"
  }
}
```

user:deactivated

An existing user’s account is disabled. This action can only be performed by a company admin on another user, or by a Lever employee for a support request.

```
"meta": {
  "user": {
    "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
    "email": "nnewton@brickly.com",
    "name": "Nelly Newton",
    "role": "admin"
  }
}
```

user:merged

Two existing user accounts are merged together. The target of the event is the canonical user, while the “duplicate” user is the one which gets merged into the other user and then deleted (generating a separate `user:removed` event for that user). This can only be done by a Lever employee for a support request.

```
"meta": {
  "user": {
    "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
    "email": "nnewton@brickly.com",
    "name": "Nelly Newton",
    "role": "admin"
  },
  "duplicate": {
    "id": "9f773786-3187-414b-97d3-49460d6e3e44",
    "email": "nelly.newton@brickly.com",
    "name": "nelly.newton",
    "role": "interviewer"
  }
}
```

user:reactivated

A deactivated user’s account is re-enabled. This action can only be performed by a company admin on another user, or by a Lever employee for a support request.

```
"meta": {
  "user": {
    "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
    "email": "nnewton@brickly.com",
    "name": "Nelly Newton",
    "role": "admin"
  }
}
```

user:removed

A user account is entirely deleted. This can only be done after a user merge by a Lever employee for a support request.

```
  "meta": {
    "user": {
      "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
      "email": "nnewton@brickly.com",
      "name": "Nelly Newton",
      "role": "admin"
    }
  }
```

user.authentication:failed

A user attempted to log in via the method configured for the account but was rejected for some reason.

```
  "meta": {
    "authentication": {
      "method": "direct",
      "error": {
        "message": "Looks like you don't have permission to access this
        "type": "user-deactivated"
      }
    },
    "user": {
      "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
      "email": "nelly@brickly.com",
      "name": "Nelly Newton",
      "role": "admin"
    }
  }
```

If there’s an attempt to sign in for a non-existent user, then the user in `user`, `target`, and `meta` may not have an ID.

```
  "meta": {
    "user": {
      "email": "nully@brickly.com"
    },
    "authentication": {
      "method": "direct",
      "error": {
        "message": "Looks like you don't have permission to access this
        "type": "user-not-found"
      }
    }
  }
```

user.authentication:succeeded

A user successfully logged in via the method configured for the account. The authentication methods are `"googleApps"`, `"saml"`, `"microsoft"`, and `"direct"` (using passwords).

```
"meta": {
  "user": {
    "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
    "email": "nelly@brickly.com",
    "name": "Nelly Newton",
    "role": "admin"
  },
  "authentication": {
    "method": "direct"
  }
}
```

user.email:changed

An existing user’s email address was changed. This can be caused by a user changing their own email, a company admin changing another user’s email, or by a Lever employee for a support request.

```
"meta": {
  "user": {
    "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
    "email": "nnewton@brickly.com",
    "name": "Nelly Newton",
    "role": "admin"
  },
  "from": "nelly@brickly.com",
  "to": "nnewton@brickly.com"
}
```

user.linkedProfile:changed

A user’s linked profile was changed. This occurs whenever a new or existing user is linked to their candidate profile.

```
"meta": {
  "user": {
    "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
    "email": "nnewton@brickly.com",
    "name": "Nelly Newton",
    "role": "admin"
  },
  "from": {
    "id": "0a2ecd8e-352f-da32-99a1-aaaee88913a"
  },
  "to": {
    "id": "aa32b022-737e-1112-9a03-9930abdd299"
  }
}
```

user.password:forgot

A user requested a password reset.

```
"meta": {
  "user": {
    "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
    "email": "nnewton@brickly.com",
    "name": "Nelly Newton",
    "role": "admin"
  }
}
```

user.password:changed

A user changed their password after a password reset or from their own user settings. The `"from"` and `"to"` values are always redacted and stored as `"****"` because they're sensitive data. (Redacted properties make it easier to display `:changed` events in a consistent way.)

```
"meta": {
  "user": {
    "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
    "email": "nnewton@brickly.com",
    "name": "Nelly Newton",
    "role": "admin"
  }
}
```

user.permission:changed

A change in the configuration of a permission that a user has. Learn more

```
"meta": {
  "user": {
    "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
    "email": "nelly@brickly.com",
    "name": "Nelly Newton",
    "role": "admin"
  },
  "from": {
    "level": "all",
    "name": "profiles_sensitive",
    "conditions": null
  },
  "to": {
    "level": "conditional",
    "name": "profiles_sensitive",
    "conditions": {
      "postingConditions": [
        {
          "department": "Engineering",
          "location": "San Francisco"
        }
      ],
      "postingIds": [
        "dd2384952-012a-22a2-0d0s-22cda2fa850"
      ]
    }
  }
}
```

user.role:changed

An existing user’s access was changed. This can happen when a company admin changes another user’s access or when a Lever employee modifies a user's access for a support request.

```
"meta": {
  "user": {
    "id": "13279cd8-f406-432d-8e8a-bb7e7afa850",
    "email": "nelly@brickly.com",
    "name": "Nelly Newton",
    "role": "super admin"
  },
  "from": "admin",
  "to": "super admin"
}
```

Candidates

Candidates are individuals who have been added to your Lever account as potential fits for your open job positions. Candidates enter your pipeline by applying to a posting on your jobs site, being added by an external recruiting agency, or being sourced from an online profile. Candidate profiles contain personal information as well as notes, feedback, interview schedules, and additional forms. Candidates exit your pipeline by being archived for one of two reasons: (1) The candidate was rejected, or (2) The candidate was hired.

Attributes

id

`c0950a08-e88e-4435-aada-ce8119a819ad`

String

Candidate UID

name

`Shane Stiles`

String

Candidate full name

headline

`Threadbird LLC, Cogistics, Inc, Warm Fuzzy Publicity`

String

Candidate headline, typically a list of previous companies where the candidate has worked or schools that the candidate has attended.

stage

`7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5`

Stage UID

The stage ID of the candidate's current stage.

Expandable If expanded, contains a stage object.

stageChanges

Array

An array of a candidate's historical stage changes

＋ Show child fields

location

`Lakeland`

String

Candidate current location

phones

`[{"value":"(123) 456-7891"}]`

Array of objects

Candidate phone number(s)

+ Show child fields

emails

`shane@gmail.com`

Array of strings

Candidate emails

links

`indeed.com/r/Shane-Stiles/0a7c87f6b246d1ac`

Array of strings

List of candidate links (e.g. personal website, LinkedIn profile, etc.)

archived

Object

Candidate archived status.

+ Show child fields

tags

`["San Francisco","Full-time","Support","Customer Success","Customer Success Manager"]`

Array

An array containing a list of tags for the candidate. Tags are specified as strings, identical to the ones displayed in the Lever interface.

sources

`["Job site"]`

Array

An array of candidate source strings.

origin

`applied`

String

The way a candidate was added to Lever

owner

`1804ed98-42e4-498b-bb2a-10a8e8f86610`

User UID

The user ID of the candidate owner.

Expandable If expanded, contains a user object.

followers

`1804ed98-42e4-498b-bb2a-10a8e8f86610,d2b18d51-d4a0-4976-a614-9e481577892a,89edbd66-6449-439a-97ee-e798424179e1`

Array

An array of user IDs of the followers of this candidate.

Expandable If expanded, contains an array of user objects

applications

`8fe832e2-46f9-4aa9-bd8d-e9d78b306d05`

Array

An array of candidate application IDs (includes both active and archived applications)

resume

ℹ WARNING: This field is deprecated. Use the resumes endpoint.

createdAt

`1407460071043`

Timestamp [?]

Datetime when candidate was created in Lever. For candidates who applied to a job posting on your website, the date and time when the candidate profile was created in Lever is the moment when the candidate clicked the "Apply" button on their application.

lastInteractionAt

`1417588008760`

Timestamp [?]

Datetime when the last interaction with the candidate profile occurred. [?]

lastAdvancedAt

`1417587916150`

Timestamp [?]

Datetime when the candidate advanced to the pipeline stage where they are currently located in your hiring process

snoozedUntil

`1505971500000`

Timestamp [?]

If candidate is snoozed, the timestamp will reflect the datetime when the snooze period ends

urls

Object

An object containing the list and show urls for the candidate.

+ Show child fields

dataProtection

Object

An object containing a candidate's data protection status based on candidate-provided consent and applicable data policy regulations. If there is no policy in place or if no policies apply to the candidate, value is `null`.

+ Show child fields

isAnonymized

`false`

Boolean

Indicates whether a candidate has been anonymized. Anonymized candidates have their personal information removed. Non-personal metadata may remain for accurate reporting purposes.

Retrieve a single candidate

```
GET /candidates/:candidate
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": {
    "id": "c0950a08-e88e-4435-aada-ce8119a819ad",
    "name": "Shane Stiles",
    "headline": "Threadbird LLC, Cogistics, Inc, Warm Fuzzy Publicity"
    "emails": [
      "shane@gmail.com"
    ],
    "phones": [
      {
        "value": "(123) 456-7891"
      }
    ],
    "location": "Lakeland",
    "links": [
      "indeed.com/r/Shane-Stiles/0a7c87f6b246d1ac"
    ],
    "createdAt": 1407460071043,
    "lastInteractionAt": 1417588008760,
    "lastAdvancedAt": 1417587916150,
    "snoozedUntil": 1505971500000,
    "archivedAt": null,
    "archiveReason": null,
    "stage": "7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5",
    "stageChanges": [
      {
        "toStageId": "7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5",
        "toStageIndex": 1,
        "userId": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
        "updatedAt": 1407460071043
      }
    ],
    "owner": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
    "tags": [
      "San Francisco",
      "Full-time",
      "Support",
      "Customer Success",
      "Customer Success Manager"
    ],
    "sources": [
      "Job site"
    ],
    "origin": "applied",
    "applications": [
      "8fe832e2-46f9-4aa9-bd8d-e9d78b306d05"
    ],
    "resume": null,
    "followers": [
      "1804ed98-42e4-498b-bb2a-10a8e8f86610",
      "d2b18d51-d4a0-4976-a614-9e481577892a",
      "89edbd66-6449-439a-97ee-e798424179e1"
    ],
    "urls": {
      "list": "https://hire.lever.co/candidates",
      "show": "https://hire.lever.co/candidates/1804ed98-42e4-498b-bb2
    },
    "dataProtection": {
      "store": {
        "allowed": true,
        "expiresAt": 1522540800000
      },
      "contact": {
        "allowed": false,
        "expiresAt": null
      }
    },
    "isAnonymized": false
  }
}
```

List all candidates

Lists all pipeline candidates in your Lever account.

```
GET /candidates
```

Parameters

include
`followers`
Optional
Include candidate followers in list results

expand
`stage, owner, followers`
Optional
Expand stage and user IDs into full objects in response

tag
`San Francisco`
Optional
Filter candidates by tag (case sensitive). Results will include candidates that contain the specified tag. Multiple tags can be specified and results will include a union of result sets (i.e. candidates that have either tag).

origin
`applied`
Optional
Filter candidates by origin. Results will include candidates that contain the specified origin. Multiple origins can be specified and results will include a union of result sets (i.e. candidates from either origin).

source_id
Optional
Filter candidates by source. Results will include candidates that contain the specified source tag. Multiple sources can be specified and results will include a union of result sets (i.e. candidates from either source).

stage_id
`7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5`
Optional
Filter candidates by current stage. Results will include candidates that are currently in the specified stage. Multiple stages can be specified and results will include a union of result sets (i.e. candidates that are in either stage).

posting_id
`1a6a7715-2d20-4309-9475-2f27cac74afa`
Optional
Filter candidates by posting. Results will include candidates that are applied to the specified posting. Multiple postings can be specified and results will include a union of result sets (i.e. candidates that are applied to either posting).

archived_posting_id
`1a6a7715-2d20-4309-9475-2f27cac74afa`
Optional
Filter candidates by postings for which they have been archived. Results will include candidates that applied to the specified posting and then the candidate or the application was archived. Multiple postings can be specified and results will include a union of result sets (i.e. candidates that were applied to either posting).

created_at_start,
created_at_end
`1407460069499`
Optional

Filter candidates by the timestamp they were created. If only created_at_start is specified, all candidates created from that timestamp (inclusive) to the present will be included. If only created_at_end is specified, all candidates created before that timestamp (inclusive) are included.

updated_at_start, updated_at_end

`1407460069499`

Optional

Filter candidates by the timestamp they were last updated. If only updated_at_start is specified, all candidates updated from that timestamp (inclusive) to the present will be included. If only updated_at_end is specified, all candidates updated before that timestamp (inclusive) are included.

advanced_at_start, advanced_at_end

`1407460069499`

Optional

Filter candidates by the timestamp they were advanced to their current stage. If only advanced_at_start is specified, all candidates advanced from that timestamp (inclusive) to the present will be included. If only advanced_at_end is specified, all candidates advanced before that timestamp (inclusive) are included.

archived_at_start, archived_at_end

`1407460069499`

Optional

Filter candidates by the timestamp they were archived. If only archived_at_start is specified, all candidates archived from that timestamp (inclusive) to the present will be included. If only archived_at_end is specified, all candidates archived before that timestamp (inclusive) are included.

archived

`true`

Optional

Filter candidates by archive status. If unspecified, results include both archived and unarchived candidates. If true, results only include archived candidates. If false, results only include active candidates.

archive_reason_id

`141b14d3-1d1a-4564-9b87-d0b3d12dd948`

Optional

Filter candidates by archive reason. Results will include candidates that are have been archived with the specified reason. Multiple archive reasons can be specified and results will include a union of result sets (i.e. candidates that have been archived for either reason).

snoozed

`true`

Optional

Filter candidates by snoozed status. If unspecified, results include both snoozed and unsnoozed candidates. If true, results only include snoozed candidates. If false, results only include unsnoozed candidates.

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates
```

```
{
  "data": [
{
  "id": "c0950a08-e88e-4435-aada-ce8119a819ad",
  "name": "Shane Stiles",
  "headline": "Threadbird LLC, Cogistics, Inc, Warm Fuzzy Publicit
  "emails": [
    "shane@gmail.com"
  ],
  "phones": [
    {
      "value": "(123) 456-7891"
    }
  ],
  "location": "Lakeland",
  "links": [
    "indeed.com/r/Shane-Stiles/0a7c87f6b246d1ac"
  ],
  "createdAt": 1407460071043,
  "lastInteractionAt": 1417588008760,
  "lastAdvancedAt": 1417587916150,
  "snoozedUntil": 1505971500000,
  "archivedAt": null,
  "archiveReason": null,
  "stage": "7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5",
  "stageChanges": [
    {
      "toStageId": "7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5",
      "toStageIndex": 1,
      "userId": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
      "updatedAt": 1407460071043
    }
  ],
  "owner": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
  "tags": [
    "San Francisco",
    "Full-time",
    "Support",
    "Customer Success",
    "Customer Success Manager"
  ],
  "sources": [
    "Job site"
  ],
  "origin": "applied",
  "applications": [
    "8fe832e2-46f9-4aa9-bd8d-e9d78b306d05"
  ],
  "resume": null,
  "followers": [
    "1804ed98-42e4-498b-bb2a-10a8e8f86610",
    "d2b18d51-d4a0-4976-a614-9e481577892a",
    "89edbd66-6449-439a-97ee-e798424179e1"
  ],
  "urls": {
    "list": "https://hire.lever.co/candidates",
    "show": "https://hire.lever.co/candidates/1804ed98-42e4-498b-b
  },
  "dataProtection": {
    "store": {
      "allowed": true,
      "expiresAt": 1522540800000
    },
    "contact": {
      "allowed": false,
      "expiresAt": null
    }
  },
  "isAnonymized": false
},
{
  "id": "d92ec133-876f-4d00-9390-8a3e933be1cb",
  "name": "Yifan Wu",
  "headline": "DTN Management Co, Yongda Jaguar-Land Rover Auto Sa
  "emails": [
    "yifan@gmail.com"
  ],
  "phones": [
    {
      "value": "(123) 456-7891"
    }
  ],
  "location": "Brighton",
  "links": [
    "indeed.com/r/Yifan-Wu/4f2c7523b0edefaa"
  ],
  "createdAt": 1407778275799,
  "lastInteractionAt": 1417587990376,
  "lastAdvancedAt": 1417587903121,
  "snoozedUntil": 1499577840000,
  "archivedAt": null,
  "archiveReason": null,
  "stage": "7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5",
  "owner": "d2b18d51-d4a0-4976-a614-9e481577892a",
  "tags": [
    "San Francisco",
    "Marketing",
    "Customer Success",
    "Customer Success Manager",
    "Full-time"
  ],
  "sources": [
    "Job site"
  ],
  "origin": "applied",
  "applications": [
    "a51ec8ee-03b8-4b44-a4bd-2970729f9c34"
  ],
  "resume": null,
  "followers": [
    "d2b18d51-d4a0-4976-a614-9e481577892a",
    "d2b18d51-d4a0-4976-a614-9e481577892a",
    "89edbd66-6449-439a-97ee-e798424179e1"
  ],
  "dataProtection": null
},
{
  "id": "b90ea5c0-8a6a-4782-83e7-693447778689",
  "name": "Robert Euerle",
  "headline": "Visual Information Access",
  "emails": [
    "robert@gmail.com"
  ],
  "phones": [
    {
      "value": "(123) 456-7891"
    }
  ],
  "location": "Boston",
  "links": [
    "http://linkedin.com/in/roberte"
  ],
  "createdAt": 1407778277088,
  "lastInteractionAt": 1417587981210,
      "lastAdvancedAt": 1417587891220,
      "snoozedUntil": 1420266291216,
      "archivedAt": null,
      "archiveReason": null,
      "stage": "7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5",
      "owner": "d2b18d51-d4a0-4976-a614-9e481577892a",
      "tags": [
        "San Francisco",
        "Marketing",
        "Customer Success",
        "Customer Success Manager",
        "Full-time"
      ],
      "sources": [
        "Job site"
      ],
      "applications": [
        "48f27a18-a117-4ad0-837b-42488743d45f"
      ],
      "resume": null,
      "followers": [
        "d2b18d51-d4a0-4976-a614-9e481577892a",
        "d2b18d51-d4a0-4976-a614-9e481577892a",
        "89edbd66-6449-439a-97ee-e798424179e1"
      ],
      "dataProtection": null
    }
  ]
}
```

Create a candidate

```
POST /candidates
```

This endpoint enables integrations to create candidates in your Lever account.

If you want to apply a candidate to a job posting or create a custom job site, you should use the Lever Postings API instead of the Lever Data API.

We accept requests of type `application/json` and `multipart/form-data`. If you are including a resume or other files, you must use the `multipart/form-data` type.

There are many ways to create a candidate. Here are some examples:

- Provide a `JSON` representation of a candidate with basic information like candidate name, email, and phone number
- Upload just a resume file and specify you'd like the resume parsed. Information parsed from the resume will be used to create the candidate—their name, email, and phone number, for example.

Note: If candidate is provided in the `POST` request and resume parsing is requested, the manually provided information will always take precedence over the parsed information from the candidate resume.

All fields are optional, but an empty candidate is not particularly interesting. All query parameters except the `perform_as` parameter are optional.

Parameters

Specify query parameters in the url (e.g. `POST /candidates?perform_as=d2b18d51-d4a0-4976-a614-9e481577892a`).

perform_as

`d2b18d51-d4a0-4976-a614-9e481577892a`

Required

Perform this create on behalf of a specified user. The creator and the owner of this candidate will default to the `perform_as` user. The `owner` can be explicitly specified in the request body if you want the owner to be a different person.

parse

`true`

Optional

If unspecified, assumed to be `false`. If set to `true` and a resume file is provided, the resume will be parsed and extracted data will be used to autofill information about the candidate such as email and phone number. Any fields manually passed to the endpoint take precedence over any parsed data.

dedupe

`true`

Optional

If unspecified, assumed to be `false`. If set to `true` and an email address is provided, we will attempt to dedupe the candidate and merge provided data into the existing profile.

If the existing candidate is archived as "Hired" we always create a new candidate (i.e. we never merge a new candidate into a hired candidate).

If both candidates are active, we simply merge the candidate data into one candidate.

If both candidates are archived, we keep the most recent candidate's archive reason.

If the existing candidate is archived and the new candidate is active, we unarchive the candidate and move merged candidate to the new candidate's specified stage (defaults to New Lead).

If the existing candidate is active and the new candidate is archived, we merge the candidate data and keep the candidate in the active state.

perform_as_posting_owner

`true`

Optional

If unspecified, assumed to be `false` and the candidate owner will default to the perform_as user. If set to `true`, an array of posting IDs must be passed in via the postings field. The candidate owner will be set to that of the first posting owner. If the posting does not have an owner, the candidate owner will default to the perform_as user.

Fields

If a resume file is provided and parsing is enabled, the following fields can be populated with extracted data (although no information is guaranteed to be extracted): `name`, `headline`, `emails`, `phones`, `location`, `company`, and `links`. The candidate `headline` is typically composed of prior companies where they worked or schools that they attended.

When using the `multipart/form-data` encoding, specify each array fields individually. For example, for emails specify a `emails[]` field for each email address.

For any fields with nested values, you can specify the values of nest fields using the syntax `archive[archivedAt]=1407460071043`.

---

name
`Shane Stiles`
String
Candidate full name

headline
`Threadbird LLC, Cogistics, Inc, Warm Fuzzy Publicity`
String
Candidate headline, typically a list of previous companies where the candidate has worked or schools that the candidate has attended. This field can also be populated by parsing a provided resume file.

stage
`7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5`
Stage UID
The stage ID of the candidate's current stage. If omitted, the candidate will be placed into the "New Lead" stage.

location
`Lakeland`
String
Candidate current location

phones
`[{"value":"(123) 456-7891"}]`
Array of objects
Optional
Candidate phone number(s)

+ Show child fields

emails
`shane@gmail.com`
Array of strings
Candidate emails

links
`indeed.com/r/Shane-Stiles/0a7c87f6b246d1ac`
Array of strings
List of candidate links (e.g. personal website, LinkedIn profile, etc.) Should be specified as a JSON array for `text/json` and as an array field `links[]` for `multipart/form-data` encoding.

tags
`["San Francisco","Full-time","Support","Customer Success","Customer Success Manager"]`
Array of strings
An array containing a list of tags to apply to the candidate. Tags are specified as strings, identical to the ones displayed in the Lever interface. If you specify a tag that does not exist yet, it will be created.

sources
`["Job site"]`
Array of strings
An array containing a list of sources to apply to the candidate. Sources are specified as strings, identical to the ones displayed in the Lever interface. If you specify a source that does not exist yet, it will be created.

origin
`applied`
String
The way a candidate was added to Lever

owner

`1804ed98-42e4-498b-bb2a-10a8e8f86610`

User UID

The candidate's owner. If not specified, candidate owner defaults to the `perform_as` user.

followers

`1804ed98-42e4-498b-bb2a-10a8e8f86610,d2b18d51-d4a0-4976-a614-9e481577892a,89edbd66-6449-439a-97ee-e798424179e1`

Array of user UIDs

An array of user IDs that should be added as followers to this candidate. The candidate creator will always be added as a follower.

resumeFile

[binary file]

Candidate resume file. Additional files can be uploaded using the `file` field, but only one resume file may be specified. Only supported in `multipart/form-data` requests.

files

[binary file]

File(s) relating to the candidate. If uploading multiple files, specify the field as an array instead (use `files[]` as the field name). Only supported in `multipart/form-data` requests.

postings

`["1a6a7715-2d20-4309-9475-2f27cac74afa"]`

Array of Posting UIDs

Each posting specified will be used to create a manually added application that will be added to this candidate.

createdAt

Timestamp [?]

To create a historical candidate, set the create time for a candidate in the past. Default is current datetime. Note that time travel in the other direction is not permitted; you cannot create a candidate in the future.

archived

Object

Optional

Candidate archived status. You must specify this field if you would like the created candidate to be archived. This is useful if you'd like to import historical data into Lever and the candidates you are creating are not active. The archive `reason` must be specified for an archived candidate (if you just set the `archivedAt` we will just ignore it). If you just specify an archive reason, `archivedAt` defaults to the current datetime. If you specify an `archivedAt` datetime, you must specify a `createdAt` datetime that occurs before `archivedAt`.

+ Show child fields

Update candidate stage

Change a candidate's current stage

```
PUT /candidates/:candidate/stage
```

Parameters

perform_as `d2b18d51-d4a0-4976-a614-9e481577892a`
Required
Perform this update on behalf of a specified user.

Fields

stage `7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5`
Stage UID
The stage ID of the candidate's current stage.

Update candidate archived state

Update a candidate's archived state. If a candidate is already archived, their archive reason can be changed or if `null` is specified as the reason, they will be unarchived. If a candidate is active, they will be archived with the reason provided.

```
PUT /candidates/:candidate/archived
```

Parameters

perform_as `d2b18d51-d4a0-4976-a614-9e481577892a`
Required
Perform this update on behalf of a specified user.

Fields

reason `141b14d3-1d1a-4564-9b87-d0b3d12dd948`
Archive reason UID
Reason why this candidate is archived

cleanInterviews `true`
Boolean
Remove pending interviews from candidate when they are archived. If unspecified, defaults to `false`.

Update candidate tags

Add tags to a candidate

```
POST /candidates/:candidate/addTags
```

Remove tags from a candidate

```
POST /candidates/:candidate/removeTags
```

Parameters

perform_as	`d2b18d51-d4a0-4976-a614-9e481577892a` Required Perform this update on behalf of a specified user.

Fields

tags	`["Infrastructure Engineer"]` Array of strings Array of tags to add or remove from the candidate.

Update candidate sources

Add sources to a candidate

```
POST /candidates/:candidate/addSources
```

Remove sources from a candidate

```
POST /candidates/:candidate/removeSources
```

Parameters

perform_as	`d2b18d51-d4a0-4976-a614-9e481577892a` Required Perform this update on behalf of a specified user.

Fields

sources	`["Gild"]` Array of strings Array of sources to add or remove from the candidate.

Add candidate postings

Add postings to a candidate

```
POST /candidates/:candidate/addPostings
```

Parameters

perform_as

`d2b18d51-d4a0-4976-a614-9e481577892a`

Required

Perform this update on behalf of a specified user.

Fields

postings

`["1a6a7715-2d20-4309-9475-2f27cac74afa"]`

Array of Posting UIDs

Array of posting UIDs to be applied to this candidate.

Feedback

Feedback forms are added to a candidate's profile as they are completed after interviews by interviewers or they can be manually added directly to the profile. Learn more about customizing your feedback form templates in Lever.

Attributes

id

`b1891a31-a7b0-49ec-a2a3-6fc3ed9eecc4`

String

Form UID

type

`interview`

String

Form type. Feedback forms are of type `interview`.

text

`On-site interview`

String

Form title. This can be edited in Feedback and Form Settings. [?]

instructions

`Ask about goals, fears and hopes. Remember to play smooth jazz in the background.`

String

Form instructions.

baseTemplateId

`84548101-9b84-433d-8491-d2ca655a2aab`

Form Template UID

Form template UID. This form represents a completed form template.

fields

Array

An array of form fields. Feedback forms support the follow field types:

code - for programming questions
date - special field for dates
dropdown - a dropdown menu
multiple choice - choose only one
multiple select - choose 1 or more
score system - overall candidate rating
score - thumbs up / thumbs down format.
scorecard - customized evaluation for multiple skills
text - single line answer

textarea - longer form answer
yes/no - a yes or no question

user
`1804ed98-42e4-498b-bb2a-10a8e8f86610`
User UID
The user who completed and submitted the feedback.
Expandable If expanded, contains a user object.

interview
`e32fa788-9b1b-4306-8f16-c793356c2ccf`
Interview UID
The interview for which the feedback was submitted. Manually added feedback forms will not be associated with an interview.

createdAt
`1423231493557`
Timestamp [?]
Datetime when form was created.

completedAt
`1423231549510`
Timestamp [?]
Datetime when form was completed.

deletedAt
`1526925087354`
Timestamp [?]
Datetime when form was deleted.

Retrieve a feedback form

```
GET /candidates/:candidate/feedback/:feedback
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": {
    "id": "b1891a31-a7b0-49ec-a2a3-6fc3ed9eecc4",
    "type": "interview",
    "text": "On-site interview",
    "instructions": "Ask about goals, fears and hopes. Remember to pla
    "fields": [
      {
        "type": "score",
        "text": "Rating",
        "description": "",
        "required": true,
        "value": "4 - Strong Hire",
        "prompt": "Select one",
        "options": [
          {
            "text": "4 - Strong Hire"
          },
          {
            "text": "3 - Hire"
          },
          {
            "text": "2 - No Hire"
          },
          {
            "text": "1 - Strong No Hire"
          }
        ]
      },
      {
        "type": "textarea",
        "text": "Notes / Comments",
        "description": "",
        "required": true,
        "value": "Had a great conversation with Kathryn. Sounds like t
      },
      {
        "type": "text",
        "text": "Additional Information",
        "description": "",
        "required": false,
        "value": "Has 10 year of experience as an office manager for s
      },
      {
        "type": "multiple-choice",
        "text": "What is the candidate's greatest strength?",
        "description": "",
        "required": false,
        "options": [
          {
            "text": "Execution"
          },
          {
            "text": "Strategic Thinking"
          },
          {
            "text": "Relationship Building"
          },
          {
            "text": "Influencing"
          }
        ],
        "value": "Execution"
      },
      {
        "type": "dropdown",
        "text": "What is your recommended next step?",
        "description": "",
        "required": false,
        "prompt": "Please select next step",
        "options": [
          {
            "text": "Candidate needs more selling"
          },
          {
            "text": "Bring candidate in for additional interviews"
          },
          {
            "text": "Move forward with candidate"
          }
        ],
        "value": "Bring candidate in for additional interviews"
      },
      {
        "type": "multiple-select",
        "text": "What questions did you ask the candidate?",
        "description": "",
        "required": false,
        "options": [
          {
            "text": "What programming languages are you most comfortab
          },
          {
            "text": "Can you email us two or three references that we
          },
          {
            "text": "What project are you most proud of?"
          }
        ],
        "value": [
          "Can you email us two or three references that we can speak
          "What project are you most proud of?"
        ]
      }
    ],
    "baseTemplateId": "84548101-9b84-433d-8491-d2ca655a2aab",
    "interview": "e32fa788-9b1b-4306-8f16-c793356c2ccf",
    "user": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
    "createdAt": 1423231493557,
    "completedAt": 1423231549510,
    "deletedAt": 1526925087354
  }
}
```

List all feedback

Lists all feedback forms for a candidate

```
GET /candidates/:candidate/feedback
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": [
    {
      "id": "b1891a31-a7b0-49ec-a2a3-6fc3ed9eecc4",
      "type": "interview",
      "text": "On-site interview",
      "instructions": "Ask about goals, fears and hopes. Remember to p
      "fields": [
        {
          "type": "score",
          "text": "Rating",
          "description": "",
          "required": true,
          "value": "4 - Strong Hire",
          "prompt": "Select one",
          "options": [
            {
              "text": "4 - Strong Hire"
            },
            {
              "text": "3 - Hire"
            },
            {
              "text": "2 - No Hire"
            },
            {
              "text": "1 - Strong No Hire"
            }
          ]
        },
        {
          "type": "textarea",
          "text": "Notes / Comments",
          "description": "",
          "required": true,
          "value": "Had a great conversation with Kathryn. Sounds like
        },
        {
          "type": "text",
          "text": "Additional Information",
          "description": "",
          "required": false,
          "value": "Has 10 year of experience as an office manager for
        },
        {
          "type": "multiple-choice",
          "text": "What is the candidate's greatest strength?",
          "description": "",
          "required": false,
          "options": [
            {
              "text": "Execution"
            },
            {
              "text": "Strategic Thinking"
            },
            {
              "text": "Relationship Building"
            },
            {
              "text": "Influencing"
            }
          ],
          "value": "Execution"
        },
        {
          "type": "dropdown",
          "text": "What is your recommended next step?",
          "description": "",
          "required": false,
          "prompt": "Please select next step",
          "options": [
            {
              "text": "Candidate needs more selling"
            },
            {
              "text": "Bring candidate in for additional interviews"
            },
            {
              "text": "Move forward with candidate"
            }
          ],
          "value": "Bring candidate in for additional interviews"
        },
        {
          "type": "multiple-select",
          "text": "What questions did you ask the candidate?",
          "description": "",
          "required": false,
          "options": [
            {
              "text": "What programming languages are you most comfort
            },
            {
              "text": "Can you email us two or three references that w
            },
            {
              "text": "What project are you most proud of?"
            }
          ],
          "value": [
            "Can you email us two or three references that we can spea
            "What project are you most proud of?"
          ]
        }
      ],
      "baseTemplateId": "84548101-9b84-433d-8491-d2ca655a2aab",
      "interview": "e32fa788-9b1b-4306-8f16-c793356c2ccf",
      "user": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
      "createdAt": 1423231493557,
      "completedAt": 1423231549510,
      "deletedAt": 1526925087354
    }
  ],
  "hasNext": false
}
```

Feedback Templates

A feedback template is a type of form that is used to evaluate candidates.

Attributes

id `c0950a08-e88e-4435-aada-ce8119a819ad`

Feedback Template UID

Feedback Template UID

text `Phone Screen Feedback Form`

String

Name of the feedback template

group

Object

Group object that the feedback template belongs to. Users can organize feedback templates within "groups", but not all feedback templates will necessarily belong to a group. Group may be an empty string or null.

+ Show child fields

List all feedback templates

Lists all active feedback templates for an account. Including 'no-feedback' which is a default on all accounts.

```
GET /feedback_templates
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/feedback_templates
```

```
{
  "data": [
    {
      "id": "c0950a08-e88e-4435-aada-ce8119a819ad",
      "text": "Phone Screen Feedback Form",
      "group": {
        "id": "c0950a08-e88e-4435-aada-ce8119a819ad",
        "name": "Engineering Feedback Forms"
      }
    },
    {
      "id": "c0950a08-e88e-4435-aada-ce8119a819ad",
      "text": "Onsite Technical Interview Feedback Form",
      "group": {
        "id": "c0950a08-e88e-4435-aada-ce8119a819ad",
        "name": ""
      }
    },
    {
      "id": "c0950a08-e88e-4435-aada-ce8119a819ad",
      "text": "No Feedback",
      "group": null
    }
  ],
  "hasNext": false
}
```

Files

These endpoints are for files that have been uploaded to a candidate profile. If you're looking specifically for the candidates' resumes see the Resumes endpoint. Candidate files may include cover letters, portfolios, or images.

Attributes

id `e9523h21-c31c-3982-aaba-ca8392c938dd`
String
File UID

downloadUrl `https://hire.lever.co/candidates/96118ae8-3d48-4987-b7df-29d6ca86e3fa/files/e9523h21-c31c-3982-aaba-ca8392c938dd/download`
String
File download URL

ext `.pdf`
String
File extension

name `FinJaneArticle.pdf`

String

File name

uploadedAt `1468953420000`

Timestamp [?]

Datetime when file was uploaded in Lever

Retrieve a single file

This endpoint retrieves the metadata for a single file. To download a file, see the file download endpoint.

```
GET /candidates/:candidate/files/:file
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": {
    "id": "e9523h21-c31c-3982-aaba-ca8392c938dd",
    "downloadUrl": "https://hire.lever.co/candidates/96118ae8-3d48-498
    "ext": ".pdf",
    "name": "FinJaneArticle.pdf",
    "uploadedAt": "1468953420000"
  }
}
```

List all files

```
GET /candidates/:candidate/files
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
[
  {
    "id": "e9523h21-c31c-3982-aaba-ca8392c938dd",
    "downloadUrl": "https://hire.lever.co/candidates/96118ae8-3d48-498
    "ext": ".pdf",
    "name": "FinJaneArticle.pdf",
    "uploadedAt": "1468953420000"
  },
  {
    "id": "c2938b21-d22c-3982-ccab-ca3242c328ef",
    "downloadUrl": "https://hire.lever.co/candidates/96118ae8-3d48-498
    "ext": ".png",
    "name": "logo.png",
    "uploadedAt": "1469353420000"
  },
  {
    "id": "b3912e39-c31c-2312-bbad-ca8392c938dd",
    "downloadUrl": "https://hire.lever.co/candidates/96118ae8-3d48-498
    "ext": ".pdf",
    "name": "FinJaneDesignPortfolio.pdf",
    "uploadedAt": "1471253420000"
  },
  {
    "id": "f9523h11-c23d-1293-cccc-cs9391c382fd",
    "downloadUrl": "https://hire.lever.co/candidates/96118ae8-3d48-498
    "ext": ".pdf",
    "name": "FinJaneOnSite.pdf",
    "uploadedAt": "147193420000"
  }
]
```

Download a file

```
GET /candidates/:candidate/files/:file/download
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

Form Fields

These are the currently supported field types in Lever. Note that most forms (e.g. referrals, feedback) only support a subset of available field types. Visit the docs for that specific form type to see what is available.

Code

The code field allows interviewers to add the code that was written by candidates into their feedback forms. Code fields are only available on feedback forms.

Attributes

type `code`
String
Field type (e.g. text, textarea, dropdown)

text `Longest palindromic substring`
String
Field label text

description `Write a function that finds the longest palindromic substring given a string.`
String
Field description

required `true`

Boolean

Required field flag. True if the field is required. False, otherwise.

language

`java`

String

Programming language

value

```
public static String longestPalindrome1(String
s) { int maxPalinLength = 0; String
longestPalindrome = null; int length =
s.length(); // check all possible sub strings
for (int i = 0; i < length; i++) { for (int j =
i + 1; j < length; j++) { int len = j - i;
String curr = s.substring(i, j + 1); if
(isPalindrome(curr)) { if (len >
maxPalinLength) { longestPalindrome = curr;
maxPalinLength = len; } } } } return
longestPalindrome; } public static boolean
isPalindrome(String s) { for (int i = 0; i <
s.length() - 1; i++) { if (s.charAt(i) !=
s.charAt(s.length() - 1 - i)) { return false; }
} return true; }
```

String

Programming question response

Currency

A form field for collecting a monetary value of a specified currency.

Attributes

type

`currency`

String

Field type (e.g. text, textarea, dropdown)

text

`Compensation`

String

Field label text

description

`Please enter only whole numbers, no commas or letters.`

String

Field description

required

`false`

Boolean

Required field flag. True if the field is required. False, otherwise.

currency

`USD`

String

The three-character ISO 4217 Currency Code specifying what currency this field captures

value

`50000`

Number

Currency amount. Numbers only, no currency symbols.

Date

A datepicker form control

Attributes

type	`date`	String	Field type (e.g. text, textarea, dropdown)
text	`Start date`	String	Field label text
description	`Please enter anticipated new hire start date.`	String	Field description
required	`false`	Boolean	Required field flag. True if the field is required. False, otherwise.
value	`1418716800000`	Timestamp [?]	Datetime

Dropdown

The dropdown field represents a form control that presents a menu of options. The options within the menu are specified in the `options` array.

Attributes

type	`dropdown`	String	Field type (e.g. text, textarea, dropdown)
text	`Relationship`	String	Field label text
description		String	Field description
required	`true`	Boolean	Required field flag. True if the field is required. False, otherwise.
options	`[{"text":"Former colleague"},{"text":"Friend"},{"text":"Reputation"},{"text":"Other"},{"text":"Don't know this person"}]`	Array	An array of valid values for this form control

prompt `Select one`
String
When no option is selected, this prompt is displayed.

value `Reputation`
String
Selected dropdown value. String must match exactly with one of the strings in the options array.

File Upload

A file upload field

Attributes

description `Please upload a cover letter.`
String
Field description

required `true`
Boolean
Required field flag. True if the field is required. False, otherwise.

text `Cover letter`
String
Field label text

type `file-upload`
String
Field type (e.g. text, textarea, dropdown)

value
Object
File object

+ Show child fields

Multiple Choice

A form control with multiple options that allows only one option to be selected, more colloquially known as radio buttons.

Attributes

type `multiple-choice`
String
Field type (e.g. text, textarea, dropdown)

text `What is your favorite primary color?`
String
Field label text

description `Please select one.`
String
Field description

required
Boolean
Required field flag. True if the field is required. False, otherwise.

options `[{"text":"blue"},{"text":"yellow"},{"text":"red"}]`
Array
An array of valid values for this form control

value `blue`
String
Selected value. String must match exactly with one of the strings in the options array.

Multiple Select

A form control with multiple options that allows multiple to be selected, more colloquially known as checkboxes.

Attributes

type `multiple-select`
String
Field type (e.g. text, textarea, dropdown)

text `What languages/frameworks are you comfortable with?`
String
Field label text

description
String
Field description

required `false`
Boolean
Required field flag. True if the field is required. False, otherwise.

options `[{"text":"javascript"},{"text":"node"},{"text":"derby"}]`
Array
An array of valid values for this form control

value `javascript,node`
String
An array of selected values. Strings must match exactly with one of the strings in the options array.

Note

Notes are used to communicate on a candidate's profile.

Attributes

type
`note`
String
Field type (e.g. text, textarea, dropdown)

text
`Comment`
String
Field label text

description
String
Field description

value
`Sounds great, @josh! I will send her an email this week.`
String
Note contents

user
`1804ed98-42e4-498b-bb2a-10a8e8f86610`
User UID
User who made the note on the profile

stage
`7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5`
Stage UID
Stage of candidate when note was added to profile

createdAt
`1421715919765`
Timestamp [?]
Datetime when note was created

Score System

An overall candidate rating on a 1-4 scale that appears once on every feedback form.

Attributes

type
`score-system`
String
Field type (e.g. text, textarea, dropdown)

text
`Rating`
String
Field label text

description
String
Field description

required
`true`
Boolean
Required field flag. True if the field is required. False, otherwise.

options

`[{"text":"4 - Strong Hire"},{"text":"3 - Hire"},{"text":"2 - No Hire"},{"text":"1 - Strong No Hire"}]`
Array
An array of valid values for this form control

prompt `Select one`
String
When no option is selected, this prompt is displayed.

value `3 - Hire`
String
Score value. String must match exactly with one of the strings in the options array.

Score

An individual data point typically on a skill in thumbs up / thumbs down format. Score fields are only available on feedback forms.

Attributes

type `score`
String
Field type (e.g. text, textarea, dropdown)

text `Rating`
String
Field label text

description `Score this candidate on communication skills`
String
Field description

required `false`
Boolean
Required field flag. True if the field is required. False, otherwise.

value `3`
Integer
Score value is an integer between 1 and 4. If no score is selected, then value is `null`.

Scorecard

A customized evaluation of a candidate that may involve one or more skills assessed during the interview. Scorecard fields are only available on feedback forms. Scorecards allow a specific set of skills or experiences like accountability, self-awareness, humility, etc. to be evaluated. Interviewers rate each of these skills on the thumbs scale. They can also add an additional note or comment for each score.

Attributes

type
`scorecard`
String
Field type (e.g. text, textarea, dropdown)

text
`Skills assessment`
String
Field label text

description
`Please give a score to each skill listed.`
String
Field description

required
`false`
Boolean
Required field flag. True if the field is required. False, otherwise.

scores
Array of score objects
An array of scores that compose this scorecard
+ Show child fields

value
Array of score objects
An array of objects each with a `score` and `comment` attribute.
+ Show child fields

Text

A single-line text field

Attributes

type
`text`
String
Field type (e.g. text, textarea, dropdown)

text
`Name of referrer`
String
Field label text

description
String
Field description

required
`true`
Boolean
Required field flag. True if the field is required. False, otherwise.

value
`Rachael Stedman`
String
Text field value

Textarea

A multi-line plain-text form field

Attributes

type
`textarea`
String
Field type (e.g. text, textarea, dropdown)

text
`Comments`
String
Field label text

description
String
Field description

required
`false`
Boolean
Required field flag. True if the field is required. False, otherwise.

value
`Teresa comes recommended by an old coworker of mine. Said she would be a great fit for a cross-functional role.`
String
Textarea field value

Yes-No

A logical field with a binary value

Attributes

type
`yes-no`
String
Field type (e.g. text, textarea, dropdown)

text
`Eligible for benefits?`
String
Field label text

description
String
Field description

required
`false`
Boolean
Required field flag. True if the field is required. False, otherwise.

value
`yes`
String
Value can be either `yes`, `no`, or `null`

University

A unique dropdown populated with a list of valid universities to choose from

Attributes

type	`university` String Field type (e.g. text, textarea, dropdown)
text	`Which university did you attend?` String Field label text
required	`false` Boolean Required field flag. True if the field is required. False, otherwise.
prompt	`Choose which university you attended.` String When no option is selected, this prompt is displayed.
value	`Franklin W. Olin College of Engineering` String University

Interviews

Individual interviews can be returned on this endpoint. Interviews are also contained in interview panels, and can be updated via the Panels endpoint.

Attributes

id	`e32fa788-9b1b-4306-8f16-c793356c2ccf` Interview UID Interview UID
panel	`e32346d8-9b1b-4306-8f16-c793356c2ccf` Interview Panel UID Interview Panel UID
subject	`On-site interview - Kathryn Helgerson - Office Manager` String Interview subject
note	`SCHEDULE: 6:00 - 6:30 pm - Rachael Stedman - Call - (123) 456-7891` String Interview note
interviewers	Array

Array of interviewers

+ Show child fields

timezone

`America/Los_Angeles`

String

Name of timezone in which interview was scheduled to occur.

createdAt

`1423187000000`

Timestamp [?]

Datetime when interview was created.

date

`1423188000000`

Timestamp [?]

Datetime when interview is scheduled to occur.

duration

`30`

Number

Interview duration in minutes

location

`Call - (123) 456-7891`

String

Interview location. Usually the name of a booked conference room but can also be a phone number to call.

feedbackTemplate

`84548101-9b84-433d-8491-d2ca655a2aab`

Form template UID

The feedback form template selected for this interview.

feedbackForms

`["b1891a31-a7b0-49ec-a2a3-6fc3ed9eecc4"]`

Array of feedback UIDs

The feedback forms associated with this interview

Expandable If expanded, contains an array of feedback form objects

feedbackReminder

`once`

String

Frequency of feedback reminders (i.e. once, daily, frequently, none). Defaults to 'frequently' which is every 6 hours.

user

`d2b18d51-d4a0-4976-a614-9e481577892a`

User UID

The user who created the interview.

Expandable If expanded, contains a user object.

stage

`c449d622-9e1a-4a59-bfba-546eb4181e08`

Stage UID

The stage in which the candidate resided when this interview was scheduled.

Expandable If expanded, contains a stage object.

canceledAt

`null`

Timestamp [?]

Datetime when interview was canceled. Value is `null` if interview was never canceled.

postings

`1a6a7715-2d20-4309-9475-2f27cac74afa`

Array

List of job postings that the interview is associated with

Expandable If expanded, contains a list of posting objects

gcalEventUrl

`null`

String

Link to the associated calendar event for this interview. Value is `null` if no Google calendar event was created.

Retrieve a single interview

```
GET /candidates/:candidate/interviews/:interview
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": {
    "id": "e32fa788-9b1b-4306-8f16-c793356c2ccf",
    "panel": "e32346d8-9b1b-4306-8f16-c793356c2ccf",
    "subject": "On-site interview - Kathryn Helgerson - Office Manager
    "note": "SCHEDULE:\n6:00 - 6:30 pm - Rachael Stedman - Call - (123
    "interviewers": [
      {
        "email": "rachael@exampleq3.com",
        "id": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
        "name": "Rachael Stedman"
      }
    ],
    "timezone": "America/Los_Angeles",
    "createdAt": 1423187000000,
    "date": 1423188000000,
    "duration": 30,
    "location": "Call - (123) 456-7891",
    "feedbackTemplate": "84548101-9b84-433d-8491-d2ca655a2aab",
    "feedbackForms": [
      "b1891a31-a7b0-49ec-a2a3-6fc3ed9eecc4"
    ],
    "feedbackReminder": "once",
    "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
    "stage": "c449d622-9e1a-4a59-bfba-546eb4181e08",
    "canceledAt": null,
    "postings": [
      "1a6a7715-2d20-4309-9475-2f27cac74afa"
    ],
    "gcalEventUrl": null
  }
}
```

List all interviews

Lists all interview events for a candidate

```
GET /candidates/:candidate/interviews
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": [
    {
      "id": "e32fa788-9b1b-4306-8f16-c793356c2ccf",
      "panel": "e32346d8-9b1b-4306-8f16-c793356c2ccf",
      "subject": "On-site interview - Kathryn Helgerson - Office Manag
      "note": "SCHEDULE:\n6:00 - 6:30 pm - Rachael Stedman - Call - (1
      "interviewers": [
        {
          "email": "rachael@exampleq3.com",
          "id": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
          "name": "Rachael Stedman"
        }
      ],
      "timezone": "America/Los_Angeles",
      "createdAt": 1423187000000,
      "date": 1423188000000,
      "duration": 30,
      "location": "Call - (123) 456-7891",
      "feedbackTemplate": "84548101-9b84-433d-8491-d2ca655a2aab",
      "feedbackForms": [
        "b1891a31-a7b0-49ec-a2a3-6fc3ed9eecc4"
      ],
      "feedbackReminder": "once",
      "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
      "stage": "c449d622-9e1a-4a59-bfba-546eb4181e08",
      "canceledAt": null,
      "postings": [
        "1a6a7715-2d20-4309-9475-2f27cac74afa"
      ],
      "gcalEventUrl": null
    }
  ],
  "hasNext": false
}
```

Create an interview

Create a new interview. New interviews must be created on an existing panel object. Interviews can only be created via the API on panels where externallyManaged == true.

```
POST /candidates/:candidate/interviews
```

Parameters

Specify query parameters in the url (e.g. `POST /candidates/:candidate/interviews?perform_as=d2b18d51-d4a0-4976-a614-9e481577892a`).

perform_as	`d2b18d51-d4a0-4976-a614-9e481577892a` Required

Perform this create on behalf of a specified user. The creator of this interview will default to the `perform_as` user.

Attributes

panel

`e32346d8-9b1b-4306-8f16-c793356c2ccf`

Interview Panel UID

Required

Interview Panel UID

subject

`On-site interview - Kathryn Helgerson - Office Manager`

String

Optional

Interview subject

note

`SCHEDULE: 6:00 - 6:30 pm - Rachael Stedman - Call - (123) 456-7891`

String

Optional

Interview note

interviewers

Array

Required

Array of interviewers, must contain at least one interviewer. The id is required, additional fields will be ignored.

+ Show child fields

date

`1423188000000`

Timestamp [?]

Required

Datetime when interview is scheduled to occur.

duration

`30`

Number

Required

Interview duration in minutes, minimum value is 1.

location

`Call - (123) 456-7891`

String

Optional

Interview location. Usually the name of a booked conference room but can also be a phone number to call.

feedbackTemplate

`84548101-9b84-433d-8491-d2ca655a2aab`

Form template UID

Optional

The feedback form selected for this interview. These can be obtained from the feedback template list endpoint.

feedbackReminder

`once`

String

Optional

Frequency of feedback reminders (i.e. once, daily, frequently, none) Defaults to 'frequently' which is every 6 hours.

Examples

```
curl -H "Content-Type: application/json" -X POST -u API_KEY: -d '{
  "panel": "e32346d8-9b1b-4306-8f16-c793356c2ccf",
  "subject": "On-site interview - Kathryn Helgerson - Office Manager",
  "note": "SCHEDULE:
6:00 - 6:30 pm - Rachael Stedman - Call - (123) 456-7891
",
  "interviewers": [ { id: "1804ed98-42e4-498b-bb2a-10a8e8f86610" } ],
  "date": 1423188000000,
  "duration": 30,
  "location": "Call - (123) 456-7891",
  "feedbackTemplate": "84548101-9b84-433d-8491-d2ca655a2aab",
  "feedbackReminder": "once"
}' "https://api.lever.co/v1/candidates/undefined/interviews"
```

```
201 - Created

{
  "data": {
    "data": [
      {
        "id": "e32fa788-9b1b-4306-8f16-c793356c2ccf",
        "panel": "e32346d8-9b1b-4306-8f16-c793356c2ccf",
        "subject": "On-site interview - Kathryn Helgerson - Office Mar
        "note": "SCHEDULE:\n6:00 - 6:30 pm - Rachael Stedman - Call -
        "interviewers": [
          {
            "email": "rachael@exampleq3.com",
            "id": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
            "name": "Rachael Stedman"
          }
        ],
        "timezone": "America/Los_Angeles",
        "createdAt": 1423187000000,
        "date": 1423188000000,
        "duration": 30,
        "location": "Call - (123) 456-7891",
        "feedbackTemplate": "84548101-9b84-433d-8491-d2ca655a2aab",
        "feedbackForms": [
          "b1891a31-a7b0-49ec-a2a3-6fc3ed9eecc4"
        ],
        "feedbackReminder": "once",
        "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
        "stage": "c449d622-9e1a-4a59-bfba-546eb4181e08",
        "canceledAt": null,
        "postings": [
          "1a6a7715-2d20-4309-9475-2f27cac74afa"
        ],
        "gcalEventUrl": null
      }
    ],
    "hasNext": false
  }
}
```

Update an interview

This endpoint expects the entire object to be present the put request. Missing fields will be deleted from the object. See Create endpoint for full details.

This endpoint cannot be used to update interviews that were created within the Lever application. Only interviews within panels where externallyManaged == true can be updated via the API.

```
PUT /candidates/:candidate/interviews/:interview
```

Parameters

Specify query parameters in the url (e.g. `PUT /candidates/:candidate/interviews/:interview?perform_as=d2b18d51-d4a0-4976-a614-9e481577892a`).

perform_as `d2b18d51-d4a0-4976-a614-9e481577892a`

Required

Perform this create on behalf of a specified user. The creator of this interview will default to the `perform_as` user.

Examples

```
curl -H "Content-Type: application/json" -X POST -u API_KEY: -d '{
  "panel": "e32346d8-9b1b-4306-8f16-c793356c2ccf",
  "subject": "On-site interview - Kathryn Helgerson - Office Manager",
  "note": "SCHEDULE:
6:00 - 6:30 pm - Rachael Stedman - Call - (123) 456-7891
",
  "interviewers": [ { id: "1804ed98-42e4-498b-bb2a-10a8e8f86610" } ],
  "date": 1423188000000,
  "duration": 30,
  "location": "Call - (123) 456-7891",
  "feedbackTemplate": "84548101-9b84-433d-8491-d2ca655a2aab",
  "feedbackReminder": "once"
}' "https://api.lever.co/v1/candidates/undefined/interviews/undefined"
```

```
201 - Created

{
  "data": {
    "data": [
      {
        "id": "e32fa788-9b1b-4306-8f16-c793356c2ccf",
        "panel": "e32346d8-9b1b-4306-8f16-c793356c2ccf",
        "subject": "On-site interview - Kathryn Helgerson - Office Mar
        "note": "SCHEDULE:\n6:00 - 6:30 pm - Rachael Stedman - Call -
        "interviewers": [
          {
            "email": "rachael@exampleq3.com",
            "id": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
            "name": "Rachael Stedman"
          }
        ],
        "timezone": "America/Los_Angeles",
        "createdAt": 1423187000000,
        "date": 1423188000000,
        "duration": 30,
        "location": "Call - (123) 456-7891",
        "feedbackTemplate": "84548101-9b84-433d-8491-d2ca655a2aab",
        "feedbackForms": [
          "b1891a31-a7b0-49ec-a2a3-6fc3ed9eecc4"
        ],
        "feedbackReminder": "once",
        "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
        "stage": "c449d622-9e1a-4a59-bfba-546eb4181e08",
        "canceledAt": null,
        "postings": [
          "1a6a7715-2d20-4309-9475-2f27cac74afa"
        ],
        "gcalEventUrl": null
      }
    ],
    "hasNext": false
  }
}
```

Delete an interview

This endpoint will delete an interview from a panel. Removing the last interview from a panel will delete the panel.

To reschedule an entire panel without deleting it, use the panel update endpoint. To delete a panel, use the panel delete endpoint.

This endpoint cannot be used to delete interviews that were created within the Lever application. Only interviews within panels where externallyManaged == true can be deleted via the API.

```
DELETE /candidates/:candidate/interviews/:interview
```

Parameters

Specify query parameters in the url (e.g. `DELETE /candidates/:candidate/interviews/:interview?perform_as=d2b18d51-d4a0-4976-a614-9e481577892a`).

perform_as	`d2b18d51-d4a0-4976-a614-9e481577892a` Required Perform this delete on behalf of a specified user.

Examples

```
curl -X DELETE -u API_KEY:  "https://api.lever.co/v1/candidates/undefi
```

```
    204 - No Content
```

Notes

Unlike other forms, notes do not have a base template ID because there is only one Note template and it isn't customizable.

Attributes

id
`f206891c-b1bd-49b4-bcbf-3fc54b5a58a3`
String
Form UID

text
`Note`
String
Form title. This can be edited in Feedback and Form Settings. [?]

fields
Array
An array of form fields. Notes only support the `note` field type.
text - note field

user
`d2b18d51-d4a0-4976-a614-9e481577892a`
User UID
The user who began the note thread on the candidate's profile
Expandable If expanded, contains a user object.

secret
`false`
Boolean
Whether or not this note has been made secret.

createdAt
`1420312438111`
Timestamp [?]
Datetime when form was created.

completedAt
`1420312438111`
Timestamp [?]
Datetime when form was completed.

deletedAt
`1526925087354`
Timestamp [?]
Datetime when form was deleted.

Retrieve a single note

```
GET /candidates/:candidate/notes/:note
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": {
    "id": "f206891c-b1bd-49b4-bcbf-3fc54b5a58a3",
    "text": "Note",
    "fields": [
      {
        "type": "note",
        "text": "Comment",
        "value": "@rachael We should move forward with Teresa. Can you
        "createdAt": 1420312438110,
        "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
        "score": 3,
        "stage": "7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5"
      },
      {
        "type": "note",
        "text": "Comment",
        "value": "Sounds great, @josh! I will send her an email this w
        "createdAt": 1421715919765,
        "user": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
        "score": 3,
        "stage": "7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5"
      }
    ],
    "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
    "secret": false,
    "completedAt": 1420312438111,
    "createdAt": 1420312438111,
    "deletedAt": 1526925087354
  }
}
```

List all notes

Lists all notes on a candidate profile

```
GET /candidates/:candidate/notes
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": [
    {
      "id": "f206891c-b1bd-49b4-bcbf-3fc54b5a58a3",
      "text": "Note",
      "fields": [
        {
          "type": "note",
          "text": "Comment",
          "value": "@rachael We should move forward with Teresa. Can y
          "createdAt": 1420312438110,
          "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
          "score": 3,
          "stage": "7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5"
        },
        {
          "type": "note",
          "text": "Comment",
          "value": "Sounds great, @josh! I will send her an email this
          "createdAt": 1421715919765,
          "user": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
          "score": 3,
          "stage": "7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5"
        }
      ],
      "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
      "secret": false,
      "completedAt": 1420312438111,
      "createdAt": 1420312438111,
      "deletedAt": 1526925087354
    }
  ],
  "hasNext": false
}
```

Offers

Offers capture the data sent to a candidate about a position they have been offered sent using Lever's offers feature. The status, creation date, creator, and all fields of an offer are exposed by the api.

Attributes

id

`431cadfe-3ce4-401f-84a2-c87db474656e`

String

Offer UID

createdAt

`1475540115263`

Timestamp [?]

Datetime when the offer was created in Lever.

creator

`003c2db0-eb57-4677-bc5e-00b505dfacd8`

String

The user ID of the offer creator.

Expandable If expanded, contains a user object.

status

`signed`

String

A string describing the current status of the offer. Can be one of the following values:

draft - the offer is still under construction

approval-sent - the offer needs approval

approved - the offer has been approved

sent - the offer has been sent through Lever

sent-manually - the offer has been sent to the candidate outside of Lever

opened - the candidate has opened the offer

denied - the candidate denied the offer

signed - the candidate signed the offer

fields

Array

An array of fields on the offer. Includes standard fields and all custom fields. The fields on offers are simplified versions of what appears on other forms returned by the api. Only the `text` and `value` remain along with the addition of a machine friendly identifier for easier mapping.

+ Show child fields

List all offers

```
GET /candidates/:candidate/offers
```

Parameters

expand `creator`

Optional

Expand creator ID into full object in response

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": [
    {
      "id": "431cadfe-3ce4-401f-84a2-c87db474656e",
      "createdAt": 1475540115263,
      "status": "signed",
      "creator": "003c2db0-eb57-4677-bc5e-00b505dfacd8",
      "fields": [
        {
          "text": "Job posting",
          "identifier": "job_posting",
          "value": "6b33bd36-87a1-4e23-be8b-74529c142fc2"
        },
        {
          "text": "Job title",
          "identifier": "job_title",
          "value": "Back End Software Engineer"
        },
        {
          "text": "Candidate name",
          "identifier": "candidate_name",
          "value": "Marvin Gaye"
        },
        {
          "text": "Offered compensation amount",
          "identifier": "salary_amount",
          "value": 1319000
        },
        {
          "text": "Compensation currency",
          "identifier": "compensation_currency",
          "value": "USD"
        },
        {
          "text": "Type of earnings",
          "identifier": "salary_interval",
          "value": "per-year-salary"
        },
        {
          "text": "Hiring manager",
          "identifier": "hiring_manager",
          "value": "5eca840d-3f89-48b7-b718-e3f655954959"
        },
        {
          "text": "Today's date",
          "identifier": "today_date",
          "value": 1475540150185
        },
        {
          "text": "Direct manager",
          "identifier": "direct_manager",
          "value": "55e8bf39-b169-4125-ab7b-11c7c660e2cb"
        },
        {
          "text": "Anticipated start date",
          "identifier": "anticipated_start_date",
          "value": 1475540150185
        },
        {
          "text": "equity",
          "identifier": "custom_equity",
          "value": 12345
        }
      ]
    },
    {
      "id": "f26d2f64-558f-407b-876a-0c79277550e9",
      "createdAt": 1475819157595,
      "status": "draft",
      "creator": "003c2db0-eb57-4677-bc5e-00b505dfacd8",
      "fields": [
        {
          "text": "Job posting",
          "identifier": "job_posting",
          "value": "e667cca0-5d61-4357-b048-0e5bf3e17047"
        },
        {
          "text": "Job title",
          "identifier": "job_title",
          "value": "Front End Software Engineer"
        },
        {
          "text": "Requisition",
          "identifier": "requisition",
          "value": "59eb1ba8-7e38-4d0d-bfba-6cf3fe53befa"
        },
        {
          "text": "Candidate name",
          "identifier": "candidate_name",
          "value": "Marvin Gaye"
        },
        {
          "text": "Offered compensation amount",
          "identifier": "salary_amount",
          "value": 1319000
        },
        {
          "text": "Compensation currency",
          "identifier": "compensation_currency",
          "value": "USD"
        },
        {
          "text": "Type of earnings",
          "identifier": "salary_interval",
          "value": "per-year-salary"
        },
        {
          "text": "equity",
          "identifier": "custom_equity",
          "value": 12345
        },
        {
          "text": "Today's date",
          "identifier": "today_date",
          "value": 1475540150185
        },
        {
          "text": "Direct manager",
          "identifier": "direct_manager",
          "value": "55e8bf39-b169-4125-ab7b-11c7c660e2cb"
        },
        {
          "text": "Anticipated start date",
          "identifier": "anticipated_start_date",
          "value": 1475540150185
        }
      ]
    }
  ],
  "hasNext": false
}
```

Panels

An interview panel contains a series of interviews that will take place for a given candidate in a stage.

Attributes

id `e32f1234-9b1b-4306-8f16-c793356c2aab`
Panel UID
Interview Panel UID

applications `b1891a31-a7b0-49ec-a2a3-6fc3ed9eecabcd`
Array
Array of Application UIDs that the panel is associated with

canceledAt `null`
Timestamp [?]
Datetime when panel was canceled. Value is `null` if panel was never canceled.

createdAt `1423187000000`
Timestamp [?]
Datetime when panel was created.

start `1423187000000`
Timestamp [?]
Datetime when the first interview in the panel starts.

end `1423187003000`
Timestamp [?]
Datetime when the last interview in the panel ends.

timezone `America/Los_Angeles`
String
Name of timezone in which panel was scheduled to occur.

feedbackReminder

String

Frequency of feedback reminders (i.e. once, daily, frequently, none). Defaults to 'frequently' which is every 6 hours.

user

`d2b18d51-d4a0-4976-a614-9e481577892a`

User UID

The user who created the panel.

Expandable If expanded, contains a user object.

stage

`c449d622-9e1a-4a59-bfba-546eb4181e08`

Stage UID

The stage in which the candidate resided when this panel was scheduled.

Expandable If expanded, contains a stage object.

note

`Panel for Rachael`

String

Panel note

externallyManaged

`true`

Boolean

This value is true if panels are created via the API, or via an integration.

externalUrl

`www.myinterviewpanel.com`

String

This Url links to an external entity associated with this interview.

interviews

Array

Array of interview objects. See interviews GET endpoint for details.

Retrieve a single panel

```
GET /candidates/:candidate/panels/:panel
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": {
    "id": "e32f1234-9b1b-4306-8f16-c793356c2aab",
    "applications": [
      "b1891a31-a7b0-49ec-a2a3-6fc3ed9eecabcd"
    ],
    "canceledAt": null,
    "createdAt": 1423187000000,
    "end": 1423187003000,
    "externallyManaged": true,
    "externalUrl": "www.myinterviewpanel.com",
    "interviews": [
      {
        "id": "e32fa788-9b1b-4306-8f16-c793356c2ccf",
        "date": 1423188000000,
        "duration": 30,
        "feedbackReminder": "once",
        "feedbackTemplate": "84548101-9b84-433d-8491-d2ca655a2aab",
        "interviewers": [
          {
            "email": "rachael@exampleq3.com",
            "id": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
            "name": "Rachael Stedman"
          }
        ],
        "location": "Call - (123) 456-7891",
        "note": "SCHEDULE:\n6:00 - 6:30 pm - Rachael Stedman - Call - 
        "subject": "On-site interview - Kathryn Helgerson - Office Mar
      }
    ],
    "note": "Panel for Rachael",
    "stage": "c449d622-9e1a-4a59-bfba-546eb4181e08",
    "start": 1423187000000,
    "timezone": "America/Los_Angeles",
    "user": "d2b18d51-d4a0-4976-a614-9e481577892a"
  }
}
```

List all panels

Lists all interview panels for a candidate

```
GET /candidates/:candidate/panels
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": [
    {
      "id": "e32f1234-9b1b-4306-8f16-c793356c2aab",
      "applications": [
        "b1891a31-a7b0-49ec-a2a3-6fc3ed9eecabcd"
      ],
      "canceledAt": null,
      "createdAt": 1423187000000,
      "end": 1423187003000,
      "externallyManaged": true,
      "externalUrl": "www.myinterviewpanel.com",
      "interviews": [
        {
          "id": "e32fa788-9b1b-4306-8f16-c793356c2ccf",
          "date": 1423188000000,
          "duration": 30,
          "feedbackReminder": "once",
          "feedbackTemplate": "84548101-9b84-433d-8491-d2ca655a2aab",
          "interviewers": [
            {
              "email": "rachael@exampleq3.com",
              "id": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
              "name": "Rachael Stedman"
            }
          ],
          "location": "Call - (123) 456-7891",
          "note": "SCHEDULE:\n6:00 - 6:30 pm - Rachael Stedman - Call
          "subject": "On-site interview - Kathryn Helgerson - Office M
        }
      ],
      "note": "Panel for Rachael",
      "stage": "c449d622-9e1a-4a59-bfba-546eb4181e08",
      "start": 1423187000000,
      "timezone": "America/Los_Angeles",
      "user": "d2b18d51-d4a0-4976-a614-9e481577892a"
    }
  ],
  "hasNext": false
}
```

Create a panel

```
POST /candidates/:candidate/panels
```

Parameters

Specify query parameters in the url (e.g. `POST /candidates/:candidate/panels?perform_as=d2b18d51-d4a0-4976-a614-9e481577892a`).

perform_as
`d2b18d51-d4a0-4976-a614-9e481577892a`
Required
Perform this create on behalf of a specified user. The creator of this panel will default to the `perform_as` user.

Attributes

applications
`b1891a31-a7b0-49ec-a2a3-6fc3ed9eecabcd`
Array
optional
Array of Application UIDs associated with the panel. If left blank, and the candidate has applications, this field will default to all currently active applications on the profile. Active applications can be pulled from applications list endpoint, or candidates endpoint. An empty array is invalid.

timezone
`America/Los_Angeles`
String
required
Name of timezone in which panel was scheduled to occur.

feedbackReminder
String
optional
Select frequency of feedback reminders (one of: 'once', 'daily', 'frequently' (every 6 hours), or 'none'). If not provided, this field defaults to 'daily'.

note
`Panel for Rachael`
String

optional

Panel note

externalUrl

`www.myinterviewpanel.com`

String

optional

This Url links to an external entity associated with this interview.

interviews

Array

required

Non-empty array of interview objects. See interviews GET endpoint for details.

Examples

```
curl -H "Content-Type: application/json" -X POST -u API_KEY: -d '{
  "applications": ["b1891a31-a7b0-49ec-a2a3-6fc3ed9eecabcd"]
  "timezone": "America/Los_Angeles"
  "feedbackReminder":
  "note": "Panel for Rachael"
  "externalUrl": "www.myinterviewpanel.com"
  "interviews": [{
    "subject": "On-site interview - Kathryn Helgerson - Office Manager
    "note": "SCHEDULE:
6:00 - 6:30 pm - Rachael Stedman - Call - (123) 456-7891
",
    "interviewers": [{id: "1804ed98-42e4-498b-bb2a-10a8e8f86610"}],
    "date": 1423188000000,
    "duration": 30,
    "location": "Call - (123) 456-7891",
    "feedbackTemplate": "84548101-9b84-433d-8491-d2ca655a2aab",
    "feedbackReminder": "once"
  }]
}' "https://api.lever.co/v1/candidates/undefined/panels"
```

```
201 - Created

{
  "data": {
    "data": [
      {
        "id": "e32f1234-9b1b-4306-8f16-c793356c2aab",
        "applications": [
          "b1891a31-a7b0-49ec-a2a3-6fc3ed9eecabcd"
        ],
        "canceledAt": null,
        "createdAt": 1423187000000,
        "end": 1423187003000,
        "externallyManaged": true,
        "externalUrl": "www.myinterviewpanel.com",
        "interviews": [
          {
            "id": "e32fa788-9b1b-4306-8f16-c793356c2ccf",
            "date": 1423188000000,
            "duration": 30,
            "feedbackReminder": "once",
            "feedbackTemplate": "84548101-9b84-433d-8491-d2ca655a2aab"
            "interviewers": [
              {
                "email": "rachael@exampleq3.com",
                "id": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
                "name": "Rachael Stedman"
              }
            ],
            "location": "Call - (123) 456-7891",
            "note": "SCHEDULE:\n6:00 - 6:30 pm - Rachael Stedman - Cal
            "subject": "On-site interview - Kathryn Helgerson - Office
          }
        ],
        "note": "Panel for Rachael",
        "stage": "c449d622-9e1a-4a59-bfba-546eb4181e08",
        "start": 1423187000000,
        "timezone": "America/Los_Angeles",
        "user": "d2b18d51-d4a0-4976-a614-9e481577892a"
      }
    ],
    "hasNext": false
  }
}
```

Update a panel

This endpoint expects the entire object to be present in the put request. Missing fields will be deleted from the object. See Create endpoint for full details. Only panels with externallyManaged == true can be updated via the API.

Any existing interviews in the panel should have an id field, and any new interviews should not have any id specified.

```
PUT /candidates/:candidate/panels/:panel
```

Parameters

Specify query parameters in the url (e.g. `PUT /candidates/:candidate/panels/:panel?perform_as=d2b18d51-d4a0-4976-a614-9e481577892a`).

perform_as	d2b18d51-d4a0-4976-a614-9e481577892a Required Perform this update on behalf of a specified user. The creator of this panel will default to the `perform_as` user.

Examples

```
curl -H "Content-Type: application/json" -X POST -u API_KEY: -d '{
  "applications": ["b1891a31-a7b0-49ec-a2a3-6fc3ed9eecabcd"]
  "timezone": "America/Los_Angeles"
  "feedbackReminder": ""
  "note": "Panel for Rachael"
  "externalUrl": "www.myinterviewpanel.com"
  "interviews": [{
    "subject": "On-site interview - Kathryn Helgerson - Office Manager
    "note": "SCHEDULE:
6:00 - 6:30 pm - Rachael Stedman - Call - (123) 456-7891
",
    "interviewers": [{id: "1804ed98-42e4-498b-bb2a-10a8e8f86610"}],
    "date": 1423188000000,
    "duration": 30,
    "location": "Call - (123) 456-7891",
    "feedbackTemplate": "84548101-9b84-433d-8491-d2ca655a2aab",
    "feedbackReminder": "once"
  }]
}' "https://api.lever.co/v1/candidates/undefined/panels/undefined"
```

```
201 - Created
```

```
{
  "data": {
    "data": [
      {
        "id": "e32f1234-9b1b-4306-8f16-c793356c2aab",
        "applications": [
          "b1891a31-a7b0-49ec-a2a3-6fc3ed9eecabcd"
        ],
        "canceledAt": null,
        "createdAt": 1423187000000,
        "end": 1423187003000,
        "externallyManaged": true,
        "externalUrl": "www.myinterviewpanel.com",
        "interviews": [
          {
            "id": "e32fa788-9b1b-4306-8f16-c793356c2ccf",
            "date": 1423188000000,
            "duration": 30,
            "feedbackReminder": "once",
            "feedbackTemplate": "84548101-9b84-433d-8491-d2ca655a2aab"
            "interviewers": [
              {
                "email": "rachael@exampleq3.com",
                "id": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
                "name": "Rachael Stedman"
              }
            ],
            "location": "Call - (123) 456-7891",
            "note": "SCHEDULE:\n6:00 - 6:30 pm - Rachael Stedman - Cal
            "subject": "On-site interview - Kathryn Helgerson - Office
          }
        ],
        "note": "Panel for Rachael",
        "stage": "c449d622-9e1a-4a59-bfba-546eb4181e08",
        "start": 1423187000000,
        "timezone": "America/Los_Angeles",
        "user": "d2b18d51-d4a0-4976-a614-9e481577892a"
      }
    ],
    "hasNext": false
  }
}
```

Delete a panel

This endpoint will delete a panel. Only panels with externallyManaged == true can be deleted via the API.

```
DELETE /candidates/:candidate/panels/:panel
```

Parameters

Specify query parameters in the url (e.g. `DELETE /candidates/:candidate/panels/:panel?perform_as=d2b18d51-d4a0-4976-a614-9e481577892a`).

perform_as	`d2b18d51-d4a0-4976-a614-9e481577892a` Required Perform this delete on behalf of a specified user.

Examples

```
curl -X DELETE -u API_KEY:  "https://api.lever.co/v1/candidates/undefi
```

```
204 - No Content
```

Postings

Job postings organize candidates based on the specific roles that they may fit into on your growing team. Within Lever, there are four different states of job postings: `published`, `internal`, `closed`, and `drafts`.

- `published` job postings are postings that will appear on your public careers page. These are job postings that are open to the public: candidates can apply to each of these jobs. You'll want to mark a job posting as `published` when you have an open job that you are hoping to fill, and you are actively looking for applicants.
- `internal` job postings are postings that will NOT appear on your public site. Nevertheless, you can absolutely apply candidates to these `internal` postings, and members of your team can refer friends and former colleagues to one of these `internal` job postings. Please also note that anybody with the link to the `internal` posting can apply for the job.
- `closed` job postings are postings that will NOT appear on your public jobs site, likely because they are roles that you've already filled! You'll want to keep a job posting in the `closed` status instead of deleting it so that you can easily build reports on this specific job posting. If you and your team re-open the role, you can always move a `closed` posting back to `internal` or `published`.
- `draft` job postings are postings that you are actively still putting together. `draft` postings will NOT appear on your public jobs site. Once you've finalized the description associated with this new job posting, you'll want to change the status of the posting to `published` or `internal`.

Attributes

id	`1a6a7715-2d20-4309-9475-2f27cac74afa` String Posting UID

text

`Infrastructure Engineer`

String

Title of the job posting

createdAt

`1407779365624`

Timestamp [?]

Datetime when posting was created in Lever

updatedAt

`1407779365624`

Timestamp [?]

Datetime when posting was last updated

state

`published`

String

Posting's current status. Valid states are listed here.

user

`d2b18d51-d4a0-4976-a614-9e481577892a`

User UID

The user ID of the user who created the posting.

Expandable If expanded, contains a user object.

owner

`d2b18d51-d4a0-4976-a614-9e481577892a`

User UID

The user ID of the posting owner. The posting owner is the individual who is directly responsible for managing all candidates who are applied to that role.

Expandable If expanded, contains a user object.

categories

Object

An object containing the tags of various categories.

+ Show child fields

tags

Array

An array of additional posting tags.

content

Object

Content of the job posting including any custom questions that you've built into the job application.

+ Show child fields

followers

Array

An array of user IDs of the followers of this posting.

Expandable If expanded, contains an array of user objects.

reqCode

`7381912`

String

Requisition code associated with this posting.

urls

Object

An object containing the list, show and apply urls for the job posting.Will be set to null if the posting is internal.

+ Show child fields

Retrieve a single posting

This method returns the full posting record for a single posting.

```
GET /postings/:posting
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/postings/1a6a7715-2d20-4309-9
```

```
{
  "data": {
    "id": "1a6a7715-2d20-4309-9475-2f27cac74afa",
    "text": "Infrastructure Engineer",
    "createdAt": 1407779365624,
    "updatedAt": 1407779365624,
    "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
    "owner": "d2b18d51-d4a0-4976-a614-9e481577892a",
    "categories": {
      "team": "Platform",
      "department": "Engineering",
      "location": "San Francisco",
      "commitment": "Full-time",
      "level": "Senior"
    },
    "content": {
      "description": "The Infrastructure Engineer will act as...\\nSup
      "descriptionHtml": "<div>The <u><b>Infrastructure Engineer</b></
      "lists": [
        {
          "text": "Job requirements",
          "content": "<li>Quick learner</li><li>Ambitious</li>"
        }
      ],
      "closing": "Our company is proud to be an equal opportunity work
      "closingHtml": "<div>Our <a href=\"https://www.lever.co/\" class
      "customQuestions": []
    },
    "tags": [],
    "state": "published",
    "reqCode": "7381912",
    "urls": {
      "list": "https://jobs.lever.co/example",
      "show": "https://jobs.lever.co/example/1a6a7715-2d20-4309-9475-2
      "apply": "https://jobs.lever.co/example/1a6a7715-2d20-4309-9475-
    }
  }
}
```

List all postings

Lists all postings in your Lever account. This includes all published, internal, closed, and draft postings.

```
GET /postings
```

Parameters

include
`content, followers`
String (Optional)
Include posting content or followers in list results

expand
`user, owner, followers`
String (Optional)
Expand user IDs into full objects in response

state
`published`
String (Optional)
Filter postings by state. Valid states are `published`, `internal`, `closed`, and `draft`.

group
`team`
String (Optional)
Posting results can be grouped by one of four categories: `location`, `team`, `department`, and `commitment`.

team
`Sales`
String (Optional)
Filter postings by team name (e.g. Engineering, Sales, Marketing). Since tags are case-sensitive, `Sales` will not match `sales`. Multiple teams can be specified and results will include a union of result sets (i.e. postings for either team). If your company uses departments, the same team name may occur across multiple departments.

department
`Legal`
String (Optional)
Filter postings by department name. Since tags are case-sensitive, `Legal` will not match `legal`. Multiple departments can be specified and results will include a union of result sets (i.e. postings for either department).

location
`San Francisco`
String (Optional)
Filter postings by location. Tags are case-sensitive, `San Francisco` will not match `san francisco`. Multiple locations can be specified and results will include a union of result sets (i.e. postings for either location).

commitment
`Full-time`
String (Optional)
Filter postings by work type (e.g. full-time, internship). Since tags are case-sensitive, `Full-time` will not match `full-time`. Multiple work types can be specified and results will include a union of result sets (i.e. postings of either work type).

level
`Manager`
String (Optional)
Deprecated but currently maintained for backward compatibility. Filter postings by level (e.g. junior, senior, manager). Since tags are case-sensitive, `Manager` will not match `manager`. Multiple levels can be specified and results will include a union of result sets (i.e. postings of either level).

Examples

```
curl -u API_KEY: https://api.lever.co/v1/postings
```

```
{
  "data": [
    {
      "id": "1a6a7715-2d20-4309-9475-2f27cac74afa",
      "text": "Infrastructure Engineer",
      "createdAt": 1407779365624,
      "updatedAt": 1407779365624,
      "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
      "owner": "d2b18d51-d4a0-4976-a614-9e481577892a",
      "categories": {
        "team": "Platform",
        "department": "Engineering",
        "location": "San Francisco",
        "commitment": "Full-time",
        "level": "Senior"
      },
      "content": {
        "description": "The Infrastructure Engineer will act as...\\nS
        "descriptionHtml": "<div>The <u><b>Infrastructure Engineer</b>
        "lists": [
          {
            "text": "Job requirements",
            "content": "<li>Quick learner</li><li>Ambitious</li>"
          }
        ],
        "closing": "Our company is proud to be an equal opportunity wo
        "closingHtml": "<div>Our <a href=\"https://www.lever.co/\" cla
        "customQuestions": []
      },
      "tags": [],
      "state": "published",
      "reqCode": "7381912",
      "urls": {
        "list": "https://jobs.lever.co/example",
        "show": "https://jobs.lever.co/example/1a6a7715-2d20-4309-9475
        "apply": "https://jobs.lever.co/example/1a6a7715-2d20-4309-947
      }
    },
    {
      "id": "dc8816a3-d7b9-4068-ad23-821a93a8f710",
      "text": "Customer Success Manager",
      "createdAt": 1407778984761,
      "updatedAt": 1407779122023,
      "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
      "owner": "d2b18d51-d4a0-4976-a614-9e481577892a",
      "categories": {
        "team": "Customer Success",
        "location": "San Francisco",
        "commitment": "Full-time",
        "level": null
      },
      "content": {
        "description": "",
        "descriptionHtml": "",
        "lists": [],
        "closing": "",
        "closingHtml": "",
        "customQuestions": []
      },
      "tags": [],
      "state": "published",
      "reqCode": null,
      "urls": {
        "list": "https://jobs.lever.co/example",
        "show": "https://jobs.lever.co/example/dc8816a3-d7b9-4068-ad23
        "apply": "https://jobs.lever.co/example/dc8816a3-d7b9-4068-ad2
      }
    }
  ]
}
```

Include content in list response

```
GET /postings?include=content
curl -u API_KEY: https://api.lever.co/v1/postings?include=content
```

Expand followers into user objects. Even though followers is an optional field, if expanded will automatically be included.

```
GET /postings?expand=followers
curl -u API_KEY: https://api.lever.co/v1/postings?expand=followers
```

Create a posting

```
POST /postings
```

This endpoint enables integrations to create postings in your Lever account. Creating a posting through the API will not go through the approvals chain. However, API-made postings can be created as `drafts`, and postings can still go through approvals within Lever Hire.

Lever accepts requests of type `application/json`.

Parameters

Specify query parameters in the url (e.g. `POST /postings?perform_as=d2b18d51-d4a0-4976-a614-9e481577892a`).

perform_as `d2b18d51-d4a0-4976-a614-9e481577892a`
Required
Perform this create on behalf of a specified user. The creator and the owner of this posting will default to the `perform_as` user. The `owner` can be explicitly specified in the request body if you want the owner to be a different person.

Fields

text `Infrastructure Engineer`
String
required
Title of the job posting

state `published`

String

Posting's current status. Valid states are listed here. If none is provided, `state` defaults to 'draft'.

owner `d2b18d51-d4a0-4976-a614-9e481577892a`

User UID

The user ID of the posting owner. The posting owner is the individual who is directly responsible for managing all candidates who are applied to that role. If no ID is provided, the posting owner defaults to `perform_as`.

categories

Object

An object containing the tags of various categories.

+ Show child fields

tags `engineering,high-priority`

Array

An array of additional posting tags.

content

Object

Content of the job posting.

+ Show child fields

Examples

```
curl -H "Content-Type: application/json" -X POST -u API_KEY: -d '{
  "text": "Infrastructure Engineer",
  "owner": "d2b18d51-d4a0-4976-a614-9e481577892a"
  "state": "published",
  "categories": {
    "department": "Engineering",
    "team": "Platform",
    "location": "San Francisco",
    "commitment": "Full-time"
  },
  "tags": ["engineering","high-priority"],
  "content": {
    "descriptionHtml": "<div>The <u><b>Infrastructure Engineer</b></u>
    "lists": [
      {
        "text": "Job requirements",
        "content": "<li>Quick learner</li><li>Ambitious</li>"
      }
    ],
    "closingPostingHtml": "<div>Our <a href="https://www.lever.co/">cc
  }
}' ""
```

```
201 - Created

{
  "data": {
    "id": "1a6a7715-2d20-4309-9475-2f27cac74afa",
    "text": "Infrastructure Engineer",
    "createdAt": 1407779365624,
    "updatedAt": 1407779365624,
    "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
    "owner": "d2b18d51-d4a0-4976-a614-9e481577892a",
    "categories": {
      "team": "Platform",
      "department": "Engineering",
      "location": "San Francisco",
      "commitment": "Full-time"
    },
    "content": {
      "description": "The Infrastructure Engineer will act as...\\nSup
      "descriptionHtml": "<div>The <u><b>Infrastructure Engineer</b></
      "lists": [
        {
          "text": "Job requirements",
          "content": "<li>Quick learner</li><li>Ambitious</li>"
        }
      ],
      "closing": "Our company is proud to be an equal opportunity work
      "closingHtml": "<div>Our <a href=\"https://www.lever.co/\">compa
      "customQuestions": []
    },
    "tags": [
      "engineering",
      "high-priority"
    ],
    "state": "published",
    "urls": {
      "list": "https://jobs.lever.co/example",
      "show": "https://jobs.lever.co/example/1a6a7715-2d20-4309-9475-2
      "apply": "https://jobs.lever.co/example/1a6a7715-2d20-4309-9475-
    }
  }
}
```

Update a posting

```
POST /postings/:posting
```

This endpoint enables integrations to partial-update postings in your Lever account. Only the provided attributes will be updated and other attributes will not be changed on a partial update. Updating a posting through the API will not go through the approvals chain. However, API-modified postings can be created as `drafts`, and postings can still go through approvals within Lever Hire.

Lever accepts requests of type `application/json`.

Job postings are an integral part of Lever's web application, and users are able to create and edit job postings within the Lever application. To minimize conflicts, make sure to get the most recent version before making updates.

Parameters

Specify query parameters in the url (e.g. `POST /postings/:posting?perform_as=d2b18d51-d4a0-4976-a614-9e481577892a`).

perform_as	`d2b18d51-d4a0-4976-a614-9e481577892a` Required Perform this update on behalf of a specified user.

Fields

text
`Infrastructure Engineer`
String
Title of the job posting

state
`published`
String
Posting's current status. Valid states are listed here.

categories
Object
An object containing the tags of various categories.
+ Show child fields

tags
`engineering,high-priority`
Array
An array of additional posting tags.

content
Object
Content of the job posting.
+ Show child fields

Examples

```
curl -H "Content-Type: application/json" -X POST -u API_KEY: -d '{
  "id": "1a6a7715-2d20-4309-9475-2f27cac74afa",
  "text": "Infrastructure Engineer",
  "state": "published",
  "categories": {
    "department": "Engineering",
    "team": "Platform",
    "location": "San Francisco",
    "commitment": "Full-time"
  },
  "tags": ["engineering","high-priority"],
  "content": {
    "descriptionHtml": "<div>The <u><b>Infrastructure Engineer</b></u>
    "lists": [
      {
        "text": "Job requirements",
        "content": "<li>Quick learner</li><li>Ambitious</li>"
      }
    ],
    "closingPostingHtml": "<div>Our <a href="https://www.lever.co/">cc
  }
}' "https://api.lever.co/v1/postings/1a6a7715-2d20-4309-9475-2f27cac74
```

```
201 - Created

{
  "data": {
    "id": "1a6a7715-2d20-4309-9475-2f27cac74afa",
    "text": "Infrastructure Engineer",
    "createdAt": 1407779365624,
    "updatedAt": 1407779365624,
    "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
    "owner": "d2b18d51-d4a0-4976-a614-9e481577892a",
    "categories": {
      "team": "Platform",
      "department": "Engineering",
      "location": "San Francisco",
      "commitment": "Full-time"
    },
    "content": {
      "description": "The Infrastructure Engineer will act as...\\nSup
      "descriptionHtml": "<div>The <u><b>Infrastructure Engineer</b></
      "lists": [
        {
          "text": "Job requirements",
          "content": "<li>Quick learner</li><li>Ambitious</li>"
        }
      ],
      "closing": "Our company is proud to be an equal opportunity work
      "closingHtml": "<div>Our <a href=\"https://www.lever.co/\">compa
      "customQuestions": []
    },
    "tags": [
      "engineering",
      "high-priority"
    ],
    "state": "published",
    "urls": {
      "list": "https://jobs.lever.co/example",
      "show": "https://jobs.lever.co/example/1a6a7715-2d20-4309-9475-2
      "apply": "https://jobs.lever.co/example/1a6a7715-2d20-4309-9475-
    }
  }
}
```

HTML support

Following are the definitions of the types of HTML the API supports. Only tags, attributes, and styles in the following list are valid for submission. invalid HTML will return with an error.

```
DESCRIPTION HTML

Supported tags

['div', 'p', 'span', 'br', 'b', 'i', 's', 'u', 'a']

Supported attributes

['class', 'href', 'style']

Supported classes

['postings-link']

Supported in-line styles

{
  'font-size': [
    '10px' # Small
    '14px' # Normal
    '18px' # Heading 3
    '24px' # Heading 2
    '32px' # Heading 1
  ]
}
```

```
LIST HTML

Supported tags

['li', 'b', 'i', 's', 'u', 'a']

Supported attributes

['class', 'href']

Supported classes

['postings-link']
```

Posting Forms

Posting forms can be customized and added to postings as additional questions on a job application. Posting forms do not have their own endpoint, but rather make an appearance on applications as custom questions. Learn more about adding custom questions to postings.

Attributes

type `posting`
String
Form type. Since this is a posting form, type is `posting`.

text `Software Engineer`
String
Form title. This can be edited in Feedback and Form Settings. [?]

baseTemplateId `076c3cc4-b28b-452d-b8c0-4107555aa28d`
Form Template UID
Form template UID. This form represents a completed form template.

fields
Array
An array of form fields. Posting forms support the follow field types:
dropdown - a dropdown menu
file upload - upload a file
multiple choice - choose only one
multiple select - choose 1 or more
text - single line answer
textarea - longer form answer
university - a picker for a list of valid universities

user
User UID
The user ID of the posting creator. This is not the referrer, but the Lever user that created the posting, even though they are sometimes the same person.

createdAt `1423176554644`
Timestamp [?]

Datetime when form was created.

completedAt

Timestamp [?]

Datetime when form was completed.

Profile Forms

Besides feedback forms, referral forms, and notes, there are also profile forms that contain additional information about a candidate.

Attributes

id

`59c3b921-10bf-4f9c-9097-8537f5f887b2`

String

Form UID

type

`form`

String

Form type. Profile forms are of type `form`.

text

`Offer information`

String

Form title. This can be edited in Feedback and Form Settings. [?]

instructions

`Please complete all required questions.`

String

Form instructions.

baseTemplateId

`eace3080-0be9-4b79-869a-7c4dfda2f93d`

Form Template UID

Form template UID. This form represents a completed form template.

fields

Array

An array of form fields. Feedback forms support the follow field types:

currency - special field for monetary amounts
date - special field for dates
dropdown - a dropdown menu
multiple choice - choose only one
multiple select - choose 1 or more
text - single line answer
textarea - longer form answer
yes-no - logical value field

user

`d2b18d51-d4a0-4976-a614-9e481577892a`

User UID

The user who created this profile form.

Expandable If expanded, contains a user object.

stage

`62e2491f-e5e1-4c3d-b8d0-8886e7282285`

Stage UID

Stage of the candidate at the time this form was completed.

Expandable If expanded, contains a stage object.

createdAt `1417587426973`
Timestamp [?]
Datetime when form was created.

completedAt `1417587426973`
Timestamp [?]
Datetime when form was completed.

deletedAt `1526925087354`
Timestamp [?]
Datetime when form was deleted.

Retrieve a profile form

```
GET /candidates/:candidate/forms/:form
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": {
    "id": "59c3b921-10bf-4f9c-9097-8537f5f887b2",
    "type": "form",
    "text": "Offer information",
    "instructions": "Please complete all required questions.",
    "baseTemplateId": "eace3080-0be9-4b79-869a-7c4dfda2f93d",
    "fields": [
      {
        "type": "date",
        "text": "Start date",
        "description": "",
        "required": false,
        "value": 1418716800000
      },
      {
        "type": "currency",
        "text": "Compensation",
        "description": "Please enter only whole numbers, no commas or
        "required": false,
        "value": 50000,
        "currency": "USD"
      },
      {
        "type": "yes-no",
        "text": "Eligible for benefits?",
        "description": "",
        "required": false,
        "value": "yes"
      },
      {
        "type": "text",
        "text": "Manager",
        "description": "",
        "required": false,
        "value": "Jen"
      },
      {
        "type": "textarea",
        "text": "Notes",
        "description": "",
        "required": false,
        "value": null
      }
    ],
    "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
    "stage": "62e2491f-e5e1-4c3d-b8d0-8886e7282285",
    "completedAt": 1417587426973,
    "createdAt": 1417587426973,
    "deletedAt": 1526925087354
  }
}
```

List all profile forms

List all profile forms on a candidate profile

```
GET /candidates/:candidate/forms
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "id": "59c3b921-10bf-4f9c-9097-8537f5f887b2",
  "type": "form",
  "text": "Offer information",
  "instructions": "Please complete all required questions.",
  "baseTemplateId": "eace3080-0be9-4b79-869a-7c4dfda2f93d",
  "fields": [
    {
      "type": "date",
      "text": "Start date",
      "description": "",
      "required": false,
      "value": 1418716800000
    },
    {
      "type": "currency",
      "text": "Compensation",
      "description": "Please enter only whole numbers, no commas or le
      "required": false,
      "value": 50000,
      "currency": "USD"
    },
    {
      "type": "yes-no",
      "text": "Eligible for benefits?",
      "description": "",
      "required": false,
      "value": "yes"
    },
    {
      "type": "text",
      "text": "Manager",
      "description": "",
      "required": false,
      "value": "Jen"
    },
    {
      "type": "textarea",
      "text": "Notes",
      "description": "",
      "required": false,
      "value": null
    }
  ],
  "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
  "stage": "62e2491f-e5e1-4c3d-b8d0-8886e7282285",
  "completedAt": 1417587426973,
  "createdAt": 1417587426973,
  "deletedAt": 1526925087354
}
```

Referrals

Referral forms can be created on a candidate one of two ways: manually [?] or by filling out a referral form application [?].

Attributes

id
`a50ec3b5-6d55-4589-bd04-28710aaf824b`
String
Form UID

type
`referral`
String
Form type. Since this is a referral form, type is `referral`.

text
`Referral`
String
Form title. This can be edited in Feedback and Form Settings. [?]

instructions
`Ask about goals, fears and hopes. Remember to play smooth jazz in the background.`
String
Form instructions.

baseTemplateId
`1655e26a-847e-4eb6-86b2-4658a022b536`
Form Template UID
Form template UID. This form represents a completed form template.

fields
Array
An array of form fields. Referral forms support the follow field types:
currency - special field for monetary amounts
date - special field for dates
dropdown - a dropdown menu
multiple choice - choose only one
multiple select - choose 1 or more
score - thumbs up / thumbs down format.
scorecard - customized evaluation for multiple skills

text - single line answer
textarea - longer form answer
yes/no - a yes or no question

referrer

`1804ed98-42e4-498b-bb2a-10a8e8f86610`

User UID

The user ID of the referrer. Referrals that are created manually on a profile will not have a referrer ID, but they will have a string of the referrer's name in fields (the form only requests a name to support referrer's who are not also employees).

Expandable If expanded, contains a user object.

user

`1804ed98-42e4-498b-bb2a-10a8e8f86610`

User UID

The user ID of the referral creator. This is not the referrer, but the Lever user that created the referral, even though they are sometimes the same person.

Expandable If expanded, contains a user object.

stage

Stage UID

Stage of the candidate at the time this form was completed.

Expandable If expanded, contains a stage object.

createdAt

`1422503371727`

Timestamp [?]

Datetime when form was created.

completedAt

`1422503371727`

Timestamp [?]

Datetime when form was completed.

Retrieve a single referral

```
GET /candidates/:candidate/referrals/:referral
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": {
    "id": "a50ec3b5-6d55-4589-bd04-28710aaf824b",
    "type": "referral",
    "text": "Referral",
    "instructions": "Ask about goals, fears and hopes. Remember to pla
    "fields": [
      {
        "type": "text",
        "text": "Name of referrer",
        "description": "",
        "required": true,
        "value": "Rachael Stedman"
      },
      {
        "type": "dropdown",
        "text": "Relationship",
        "description": "",
        "required": true,
        "prompt": "Select one",
        "options": [
          {
            "text": "Former colleague"
          },
          {
            "text": "Friend"
          },
          {
            "text": "Reputation"
          },
          {
            "text": "Other"
          },
          {
            "text": "Don't know this person"
          }
        ],
        "value": "Reputation"
      },
      {
        "type": "textarea",
        "text": "Notes / Comments",
        "description": "",
        "required": true,
        "value": "Teresa comes recommended by an old coworker of mine.
      }
    ],
    "baseTemplateId": "1655e26a-847e-4eb6-86b2-4658a022b536",
    "user": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
    "referrer": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
    "stage": null,
    "createdAt": 1422503371727,
    "completedAt": 1422503371727
  }
}
```

List all referrals

Lists all referrals for a candidate

```
GET /candidates/:candidate/referrals
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": [
    {
      "id": "a50ec3b5-6d55-4589-bd04-28710aaf824b",
      "type": "referral",
      "text": "Referral",
      "instructions": "Ask about goals, fears and hopes. Remember to p
      "fields": [
    {
      "type": "text",
      "text": "Name of referrer",
      "description": "",
      "required": true,
      "value": "Rachael Stedman"
    },
    {
      "type": "dropdown",
      "text": "Relationship",
      "description": "",
      "required": true,
      "prompt": "Select one",
      "options": [
        {
          "text": "Former colleague"
        },
        {
          "text": "Friend"
        },
        {
          "text": "Reputation"
        },
        {
          "text": "Other"
        },
        {
          "text": "Don't know this person"
        }
      ],
      "value": "Reputation"
    },
    {
      "type": "textarea",
      "text": "Notes / Comments",
      "description": "",
      "required": true,
      "value": "Teresa comes recommended by an old coworker of mir
    }
  ],
  "baseTemplateId": "1655e26a-847e-4eb6-86b2-4658a022b536",
  "user": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
  "referrer": "1804ed98-42e4-498b-bb2a-10a8e8f86610",
  "stage": null,
  "createdAt": 1422503371727,
  "completedAt": 1422503371727
},
{
  "id": "0c62c3fb-36d6-4b20-915d-33a6435928e4",
  "type": "referral",
  "text": "Referral",
  "instructions": "Ask about goals, fears and hopes. Remember to p
  "fields": [
    {
      "type": "text",
      "text": "Name of referrer",
      "description": "",
      "required": true,
      "value": "Josh Ackerman"
    },
    {
      "type": "dropdown",
      "text": "Relationship",
      "description": "",
      "required": true,
      "prompt": "Select one",
          "options": [
            {
              "text": "Former colleague"
            },
            {
              "text": "Friend"
            },
            {
              "text": "Reputation"
            },
            {
              "text": "Other"
            },
            {
              "text": "Don't know this person"
            }
          ],
          "value": "Reputation"
        },
        {
          "type": "textarea",
          "text": "Notes / Comments",
          "description": "",
          "required": true,
          "value": "Heard about her work online."
        }
      ],
      "baseTemplateId": "1655e26a-847e-4eb6-86b2-4658a022b536",
      "user": "d2b18d51-d4a0-4976-a614-9e481577892a",
      "referrer": "d2b18d51-d4a0-4976-a614-9e481577892a",
      "stage": null,
      "createdAt": 1418354168026,
      "completedAt": 1418354168025
    }
  ],
  "hasNext": false
}
```

Requisitions

The requisitions API allows you to create requisitions in an external tool, like an HRIS, and push them into Lever. Once in Lever, you can view and associate requisitions with job postings. To push requisitions from your HRIS to Lever, you'll want to take advantage of the write functionality of our requisitions API. To do so, a Super Admin of your account must enable API management from within the Lever Requisitions settings page. Alternatively, if you are using Lever to create and organize your requisitions, you can still read from the requisitions API endpoints.

If you'd like to push data into Lever that does not fall into the provided top-level categories, Lever supports the creation of custom fields (`customFields`) in your requisitions. When adding `customFields` onto a requisition, Lever validates the schema of those custom fields (e.g. compensation band, target hire date, hiring manager, etc.) against the requisiton field schema that exists in your account. This ensures that your data is accurately stored and translated. Your requisition field schema can be managed within Lever by an account Admin or Super Admin, or via the `requisition_field` endpoint. See below for additional details about the `requisition_fields` endpoint.

Attributes

id `c1bc2919-7f27-4f9b-a087-adfd2a90ec96`

String

Requisition UID

requisitionCode

`ENG-145`

String

The unique HRIS requisition code

backfill

`false`

Boolean

Indicates if the requisition represents a Backfill (true) or new headcount (false). If unspecified, defaults to `false`

compensationBand

Object

+ Show child fields

createdAt

`1450296000000`

Timestamp [?]

The creation datetime of the requisition

creator

`2f5s4657-321w-9880-5432-7t4j56kuo83r`

Lever User ID

Expandable If expanded, contains a user object

customFields

`"cost_center": { "city": "San Francisco", "campus_code": "2", "department": "Platform Engineering" }`

Object

A map of your custom fields, along with their values (text, date, number) or a singly nested map of values (object)

employmentStatus

`full-time`

String

The work type of the requisition.
One of: 'full-time', 'part-time', 'intern', 'contractor', 'temp-worker'

name

`Senior Software Engineer, Platform`

String

The human-readable name for the requisition.

headcountHired

`1`

Integer

The number of filled seats / openings on the requisition, or the string: 'unlimited'

headcountTotal

`10`

Integer

Total headcount allotted for this requisition

hiringManager

`fae6a208-077a-4a13-94a4-2256e418509f`

Lever User ID

Expandable If expanded, contains a user object

internalNotes

`College grad hire -- very little flexibility on salary`

String

Free form text for notes and meta-data about the requisition, visible to anyone who can see the requisition in Lever

location `San Francisco`

String

The location associated with the requisition

owner `6b9188b0-9cc7-4595-8c6e-75dfb39f8980`

Lever User ID

Expandable If expanded, contains a user object

postings `1a6a7715-2d20-4309-9475-2f27cac74afa`

Array

List of job postings that the requisition is associated with

Expandable If expanded, contains a list of posting objects

status `open`

String

The status of the requisition. Can be any of: 'open', 'onHold', 'closed', or 'draft'

team `Engineering`

String

The team/department associated with the requisition

Retrieve a single requisition

```
GET /requisitions/:requisition
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/requisitions/c1bc2919-7f27-4f
```

```
200 - OK

{
  "data": {
    "id": "c1bc2919-7f27-4f9b-a087-adfd2a90ec96",
    "requisitionCode": "ENG-145",
    "name": "Senior Software Engineer, Platform",
    "backfill": false,
    "createdAt": 1450296000000,
    "creator": "2f5s4657-321w-9880-5432-7t4j56kuo83r",
    "headcountHired": 1,
    "headcountTotal": 10,
    "status": "open",
    "hiringManager": "fae6a208-077a-4a13-94a4-2256e418509f",
    "owner": "6b9188b0-9cc7-4595-8c6e-75dfb39f8980",
    "compensationBand": {
      "currency": "USD",
      "interval": "per-year-salary",
      "min": 50000,
      "max": 80000
    },
    "employmentStatus": "full-time",
    "location": "San Francisco",
    "internalNotes": "Key hires for platform -- budget for 10, prefer
    "postings": [
      "1a6a7715-2d20-4309-9475-2f27cac74afa"
    ],
    "team": "Engineering",
    "offerIds": [
      "0b3f4728-332a-1678-9854-3h4g45jaj33b"
    ],
    "customFields": {
      "cost_center": {
        "city": "San Francisco",
        "campus_code": "2",
        "department": "Platform Engineering"
      },
      "hiring_manager": {
        "name": "Josh Ackerman",
        "HRIS_id": "ja00012",
        "lever_id": "5665c51e-475d-4e77-8aa4-b2a63bae1f2c",
        "email": "josh@lever.co"
      },
      "target_hire_date": 1453320000000
    }
  }
}
```

List all requisitions

```
GET /requisitions
```

Parameters

created_at_start, created_at_end

`1407460069499`

Optional

Filter requisitions by the datetime they were created. If only created_at_start is specified, all requisitions created from that datetime to present will be included. If only created_at_end is specified, all requisitions created before that datetime will be included.

requisition_code

`ENG-145`

Optional

Filter requisitions by non-Lever requisition code

status

`open`

Optional

Filter requisitions by status

Examples

```
curl -u API_KEY: https://api.lever.co/v1/requisitions?status=open
```

```
200 - OK

{
  "data": [
    {
      "id": "c1bc2919-7f27-4f9b-a087-adfd2a90ec96",
      "requisitionCode": "ENG-145",
      "name": "Senior Software Engineer, Platform",
      "backfill": false,
      "createdAt": 1450296000000,
      "creator": "2f5s4657-321w-9880-5432-7t4j56kuo83r",
      "headcountHired": 1,
      "headcountTotal": 10,
      "status": "open",
      "hiringManager": "fae6a208-077a-4a13-94a4-2256e418509f",
      "owner": "6b9188b0-9cc7-4595-8c6e-75dfb39f8980",
      "compensationBand": {
        "currency": "USD",
        "interval": "per-year-salary",
        "min": 50000,
        "max": 80000
      },
      "employmentStatus": "full-time",
      "location": "San Francisco",
      "internalNotes": "Key hires for platform -- budget for 10, prefe
      "postings": [
        "1a6a7715-2d20-4309-9475-2f27cac74afa"
      ],
      "team": "Engineering",
      "offerIds": [
        "0b3f4728-332a-1678-9854-3h4g45jaj33b"
      ],
      "customFields": {
        "cost_center": {
          "city": "San Francisco",
          "campus_code": "2",
          "department": "Platform Engineering"
        },
        "hiring_manager": {
          "name": "Josh Ackerman",
          "HRIS_id": "ja00012",
          "lever_id": "5665c51e-475d-4e77-8aa4-b2a63bae1f2c",
          "email": "josh@lever.co"
        },
        "target_hire_date": 1453320000000
      }
    },
    {
      "id": "994438eb-a861-4c9c-a52c-e3650ef20e12",
      "requisitionCode": "ENG-9",
      "name": "Junior Software Engineer, Platform",
      "backfill": false,
      "createdAt": 1452306348935,
      "creator": "689e4421-21c2-43d6-9245-b42b6bda27f9",
      "headcountHired": 0,
      "headcountTotal": 10,
      "owner": "5665c51e-475d-4e77-8aa4-b2a63bae1f2c",
      "status": "open",
      "hiringManager": "689e4421-21c2-43d6-9245-b42b6bda27f9",
      "compensationBand": {
        "currency": "USD",
        "interval": "per-year-salary",
        "min": 100000,
        "max": 130000
      },
      "employmentStatus": "full-time",
      "location": "New York",
      "internalNotes": "College grad hire -- very little flexibility c
      "postings": [],
      "team": "Engineering",
      "offerIds": [],
      "customFields": {
        "cost_center": {
          "city": "New York",
          "campus_code": "9",
          "department": "Platform Engineering"
        },
        "target_hire_date": 1452067200000
      }
    }
  ]
}
```

Create a requisition

See requisition fields below before attempting to create a requisition with custom fields

If you're using an HRIS, it's likely that this external tool serves as the source of truth for your requisition data. If this is indeed the case, you'll want to be able to view your requisitions in Lever and associate them with job postings, offers, and hired candidates. To do so, you'll want to set up an API connecter to push your requisition data from your HRIS into Lever.

To make POST/PUT/DELETE requests to the requisitions endpoint, your Lever account must have API-management of requisitions enabled. This can be done by a Super Admin of your account via the Lever Requisitions settings page.

If you're planning to push a requisition with custom field data from your HRIS to Lever, you'll first configure your account-level requisition fields either through Lever or by using the requisition_fields endpoint (see below for further details).

Requisitions, by default, contain a `requisitionCode` field, while their `id` field is specific to Lever. This relationship between `id` and `requisitionCode` allows you to maintain links between Lever and your external data in your HRIS. POST and PUT requests, along with a `201` status code, return the newly created data with a Lever ID back to you.

Lever accepts requests of type `application/json`.

```
POST /requisitions
```

Fields

headcountTotal
`10`
Integer
required
Total headcount on the requisition, or 'unlimited'

name
`Junior Software Engineer, Platform`
String
required
The human-readable name for the requisition

requisitionCode
`ENG-9`
String
required
The unique requisition code from your HRIS

backfill

`false`

Boolean

optional

Indicates if the requisition represents a Backfill (true) or new headcount (false).
If unspecified, defaults to `false`

compensationBand

Object

optional

+ Show child fields

createdAt

`1452306348935`

Timestamp [?]

optional

The datetime that the requisition was created
If unspecified, default is time of POST

customFields

`cost_center: { "city": "New York", "campus_code": "9", "department": "Platform Engineering" }`

Object

optional, can be configured to be required

Custom Fields are not required by default, but when they are created (via the `requisition_fields` endpoint or from the requisition settings page in Lever) they can be marked required. If included, custom fields will be validated against the requisition field schema available on your account, which you can create via the `requisition_fields` endpoint or from the requisitions settings page in Lever. The key names of a custom field must match up with the `id` property on a requisition field on your account. Custom fields can be either nested or flat. See examples for further details below.

employmentStatus

`full-time`

String

optional

The work type of the position
One of: 'full-time', 'part-time', 'intern', 'contractor', 'temp-worker'

hiringManager

`689e4421-21c2-43d6-9245-b42b6bda27f9`

String

optional

The Lever User ID of the hiring manager for the requisition
If the hiring manager isn't a Lever user, this field can be omitted.

internalNotes

`College grad hire -- very little flexibility on salary`

String

optional

Free form text for notes and meta-data about the requisition, visible to anyone who can see the requisition in Lever

location

`New York`

String

optional

The location associated with the requisition

owner

`5665c51e-475d-4e77-8aa4-b2a63bae1f2c`

String

optional

The Lever User ID of the owner of the requisition
If omitted, this field defaults to the user who performed the request

status `open`
String
optional
The status of the requisition. Can be any of: 'open', 'closed', 'onHold' or 'draft'
If unspecified, default is `open`

team `Engineering`
String
optional
The team or department associated with the requisition

Examples

```
curl -H "Content-Type: application/json" -X POST -u API_KEY: -d '{
  "requisitionCode": "ENG-9",
  "name": "Junior Software Engineer, Platform",
  "headcountTotal": 10,
  "status": "open",
  "hiringManager": "689e4421-21c2-43d6-9245-b42b6bda27f9",
  "internalNotes": "College grad hire -- very little flexibility on sa
  "owner": "5665c51e-475d-4e77-8aa4-b2a63bae1f2c"
  "compensationBand": {
    "currency": "USD",
    "interval": "per-year-salary",
    "min": 100000,
    "max": 130000
  },
  "employmentStatus": "full-time",
  "location": "New York",
  "team": "Engineering",
  "customFields": {
    "cost_center": {
      "city": "New York",
      "campus_code": "9",
      "department": "Platform Engineering"
    },
    "target_hire_date": 1452067200000
  }
}' "https://api.lever.co/v1/requisitions"
```

```
201 - Created

{
  "data": {
    "id": "994438eb-a861-4c9c-a52c-e3650ef20e12",
    "requisitionCode": "ENG-9",
    "name": "Junior Software Engineer, Platform",
    "backfill": false,
    "createdAt": 1452306348935,
    "creator": "689e4421-21c2-43d6-9245-b42b6bda27f9",
    "headcountHired": 0,
    "headcountTotal": 10,
    "owner": "5665c51e-475d-4e77-8aa4-b2a63bae1f2c",
    "status": "open",
    "hiringManager": "689e4421-21c2-43d6-9245-b42b6bda27f9",
    "compensationBand": {
      "currency": "USD",
      "interval": "per-year-salary",
      "min": 100000,
      "max": 130000
    },
    "employmentStatus": "full-time",
    "location": "New York",
    "internalNotes": "College grad hire -- very little flexibility on
    "postings": [],
    "team": "Engineering",
    "offerIds": [],
    "customFields": {
      "cost_center": {
        "city": "New York",
        "campus_code": "9",
        "department": "Platform Engineering"
      },
      "target_hire_date": 1452067200000
    }
  }
}
```

Update a requisition

When your requisitions change, be sure to update them in Lever to keep the data consistent across tools. When you PUT data, Lever expects you to send the entire resource. Every field will be overwritten by the body of the PUT request. If you don't include a field, it will be deleted or reset to its default. Be sure to include all fields you still want to be populated.

```
PUT /requisitions/:requisition
```

Examples

```
curl -H "Content-Type: application/json" -X PUT -u API_KEY: -d '{
  "requisitionCode": "ENG-19",   # updated property
  "name": "Junior Software Engineer, Platform",
  "createdAt": 1452306348935,
  "headcountTotal": 10,
  "status": "closed",            # updated property
  "hiringManager": "689e4421-21c2-43d6-9245-b42b6bda27f9",
  "compensationBand": {
    "currency": "USD",
    "interval": "per-year-salary",
    "min": 100000,
    "max": 130000
  },
  "employmentStatus": "full-time",
  "location": "New York",
  "team": "Engineering",
  "customFields": {
    "cost_center": {
      "city": "New York",
      "campus_code": "9",
      "department": "Platform Engineering"
    },
    "target_hire_date": 1452067200000
  }
}' "https://api.lever.co/v1/requisitions/994438eb-a861-4c9c-a52c-e365
```

```
201 - Created

{
  "data": {
    "id": "994438eb-a861-4c9c-a52c-e3650ef20e12",
    "requisitionCode": "ENG-19",
    "name": "Junior Software Engineer, Platform",
    "backfill": true,
    "createdAt": 1452306348935,
    "creator": "689e4421-21c2-43d6-9245-b42b6bda27f9",
    "headcountHired": 0,
    "headcountTotal": 10,
    "status": "closed",
    "hiringManager": "689e4421-21c2-43d6-9245-b42b6bda27f9",
    "compensationBand": {
      "currency": "USD",
      "interval": "per-year-salary",
      "min": 100000,
      "max": 130000
    },
    "employmentStatus": "full-time",
    "location": "New York",
    "postings": [],
    "team": "Engineering",
    "offerIds": [],
    "customFields": {
      "cost_center": {
        "city": "New York",
        "campus_code": "9",
        "department": "Platform Engineering"
      },
      "target_hire_date": 1452067200000
    }
  }
}
```

Delete a requisition

If a requisition is created by mistake, or if you need to remove a requisition for any reason from your Lever account, you'll want to use the DELETE method.

```
DELETE /requisitions/:requisition
```

Examples

```
curl -H "Content-Type: application/json" -X DELETE -u API_KEY: https:/
```

```
204 - No Content
```

Requisition fields

Requisition fields are the schema for the custom fields on your requisitions. For example, if all of your requisitions include a "Compensation band" field, before creating any requisitions that include "Compensation band", you must either create the requisition field on the account through the settings page in Lever, or POST the "Compensation band" field schema to this endpoint. After doing so, this custom field will become available across your account's requisitions. See "Retrieve a single requisition field" below for an example of a "Compensation band" field schema.

Each requisition field can have one level of nested properties as `subfields`. You can have up to 100 custom fields for your account and each custom field may have up to 10 subfields.

If you delete or modify a requisition_field from your account, any existing requisitions on your account that use the deleted/modified requisition field schema will become read-only.

Attributes

id
`cost_center`
String
Field identifier, must be snake_case, max 50 characters

text
`Cost center`
String
Human-readable field name

type
`object`
String
Type of field
options: "number", "text", "date", "object"
[if the field has subfields, it is type "object"]

subfields
Array of objects
+ Show child fields

isRequired
`true`
Boolean
Boolean designating if the field is required

Retrieve a single requisition field

```
GET /requisition_fields/:requisition_field
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/requisition_fields/cost_cente
```

```
200 - OK

{
  "data": {
    "id": "cost_center",
    "text": "Cost center",
    "type": "object",
    "isRequired": true,
    "subfields": [
      {
        "id": "city",
        "text": "City",
        "type": "text"
      },
      {
        "id": "campus_code",
        "text": "Campus code",
        "type": "number"
      },
      {
        "id": "department",
        "text": "Department",
        "type": "text"
      }
    ]
  }
}
```

List all requisition fields

Lists all requisition_fields in your Lever account

```
GET /requisition_fields
```

Parameters

isRequired	`true` Optional Filter for requisition fields that are required or specifically not required
type	`number` Optional Filter requisition_fields by top-level type

Examples

```
curl -u API_KEY: https://api.lever.co/v1/requisition_fields
```

```
200 - OK

{
  "data": [
    {
      "id": "cost_center",
      "text": "Cost center",
      "type": "object",
      "isRequired": true,
      "subfields": [
        {
          "id": "city",
          "text": "City",
          "type": "text"
        },
        {
          "id": "campus_code",
          "text": "Campus code",
          "type": "number"
        },
        {
          "id": "department",
          "text": "Department",
          "type": "text"
        }
      ]
    },
    {
      "id": "hiring_manager",
      "text": "Hiring Manager",
      "type": "object",
      "isRequired": false,
      "subfields": [
        {
          "id": "name",
          "text": "Name",
          "type": "text"
        },
        {
          "id": "HRIS_id",
          "text": "HRIS ID",
          "type": "text"
        },
        {
          "id": "email",
          "text": "Email",
          "type": "text"
        }
      ]
    }
  ]
}
```

Create a requisition field

When working with requisitions via the API, we validate custom fields against the requisition field schema established in your account settings. Using this `requisition_field` endpoint, you can create a set of requisition field schema that then become available across your account for use on any requisition you POST to the API.

Requisition fields can be one of 4 types:

- `date` - UTC datetime (milliseconds since midnight 01 January, 1970 UTC)
- `number` - integer/float
- `object` - this type of field schema also contains a `subfield` property which has one level of nested field schemas.
- `text` - string

You can have up to 100 requisition field schema on your account. Each requisition field schema can have up to 10 subfields.

We accept POSTs of type `application/json` and required fields are indicated below.

```
POST /requisition_fields
```

Fields

id
`cost_center`
String
required
Field identifier

text
`Cost center`
String
required
Human-readable field name

type
`object`
String
required
Type of field
one of : "number", "text", "date", "object"

subfields
Array of objects

+ Show child fields

isRequired
`true`
Boolean
optional
Boolean designating if the field is required

Examples

```
curl -H "Content-Type: application/json" -X POST -u API_KEY: -d '{
  "id": "cost_center",
  "text": "Cost center",
  "type": "object",
  "isRequired": true,
  "subfields": [
    {
      "id": "city",
      "text": "City",
      "type": "text"
    },
    {
      "id": "campus_code",
      "text": "Campus code",
      "type": "number"
    },
    {
      "id": "department",
      "text": "Department",
      "type": "text"
    }
  ]
}' 'https://api.lever.co/v1/requisition_fields'
```

```
201 - Created
```

```
{
  "data": {
    "id": "cost_center",
    "text": "Cost center",
    "type": "object",
    "isRequired": true,
    "subfields": [
      {
        "id": "city",
        "text": "City",
        "type": "text"
      },
      {
        "id": "campus_code",
        "text": "Campus code",
        "type": "number"
      },
      {
        "id": "department",
        "text": "Department",
        "type": "text"
      }
    ]
  }
}
```

Update a requisition field

Just like the process of updating a requisition, you'll want to POST the whole requisition_field object again, with the updated properties. Anything that you do not include will be considered deleted and will be removed. This data cannot be recovered from within Lever or via our API. If you change the type of the subfield (for example: the type of "cost_center" goes from "object" to "text"), any past requisitions that use the cost_center customField will retain their original data, and these fields will become read-only.

```
PUT /requisition_fields/:requisition_field
```

Examples

```
curl -H "Content-Type: application/json" -X PUT -u API_KEY: -d '{
    "id": "cost_center",
    "text": "Cost center",
    "type": "text",
    "isRequired": true
  }' https://api.lever.co/v1/requisition_fields/undefined
```

```
201 - Created

{
  "data": {
    "id": "cost_center",
    "text": "Cost center",
    "type": "text",
    "isRequired": true
  }
}
```

Delete a requisition field

If you need to remove a requisition field that is no longer used, you can delete requisition fields from your account by sending a DELETE request to this endpoint.

```
DELETE /requisition_fields/:requisition_field
```

Examples

```
curl -H "Content-Type: application/json" -X DELETE -u API_KEY: https:/
```

```
204 - No Content
```

Resumes

Resumes are data about a candidate's work history and/or education. Some resumes may have files associated with them. Resumes can be added to a candidate's profile in a number of ways. For example, when (1) the candidate applies to a job posting and uploads a resume (usually this will have a file associated with it), (2) when a recruiter uses the Chrome extension tool via Linkedin, Github, Twitter, etc., (3) the candidate is manually added by a recruiter or (4) a resume file is added directly to a candidate profile from within Lever.

Attributes

id `f9385g29-d21e-2391-ddab-cd8193a943ed`
String
Resume UID

createdAt `1468953420000`
Timestamp [?]
Datetime when resume was created in Lever. For candidates who applied to a job posting on your website, the date and time when the candidate's resume was created in Lever is the moment when the candidate clicked the "Attach Resume/CV" button on their application.

file
Object
An object containing resume file metadata and download url.
+ Show child fields

parsedData
Object
The candidate's parsed resume, usually extracted from an attached PDF/Word document or online profile (LinkedIn, GitHub, AngelList, etc.).
+ Show child fields

Retrieve a single resume

This endpoint retrieves the metadata for a single resume. To download a resume, see the resume download endpoint.

```
GET /candidates/:candidate/resumes/:resume
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": {
    "id": "f9385g29-d21e-2391-ddab-cd8193a943ed",
    "createdAt": 1468953420000,
    "file": {
      "name": "JulieYoungUpdated.pdf",
      "ext": "pdf",
      "downloadUrl": "https://hire.lever.co/candidates/96118ae8-3d48-4
      "uploadedAt": 1468953420004
    },
    "parsedData": {
      "positions": [
        {
          "org": "Laugh Factory",
          "title": "VP Digital Media Productions",
          "summary": "• Went from executive assistant to the CEO to ru
          "location": "Los Angeles, CA",
          "start": {
            "year": 2012,
            "month": 7
          },
          "end": {
            "year": 2016,
            "month": 3
          }
        },
        {
          "org": "Intuitive Entertainment",
          "title": "Production Assistant, Production Coordinator",
          "summary": "• PA for several reality TV shows produced by In
          "location": "Los Angeles, CA",
          "start": {
            "year": 2010,
            "month": 9
          },
          "end": {
            "year": 2012,
            "month": 3
          }
        },
        {
          "org": "Violet Pulications",
          "title": "Editor in Chief, Layout, Photographer",
          "summary": "• Editor in chief of college yearbook• Grew year
          "location": "New York, NY",
          "start": {
            "year": 2008,
            "month": 9
          },
          "end": {
            "year": 2010,
            "month": 5
          }
        }
      ],
      "schools": [
        {
          "org": "New York University Stern",
          "degree": "BS Marketing",
          "field": "Marketing, Studio Arts and Film",
          "summary": "Marketing degree with a focus on the film indust
          "start": {
            "year": 2007,
            "month": 9
          },
          "end": {
            "year": 2010,
            "month": 5
          }
        },
        {
          "org": "Savannah College of Art and Design",
          "degree": "",
          "field": "Graphic Design, Studio Arts",
          "summary": null,
          "start": {
            "year": 2006,
            "month": 9
          },
          "end": {
            "year": 2007,
            "month": 6
          }
        }
      ]
    }
  }
}
```

List all resumes

```
GET /candidates/:candidate/resumes
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

```
{
  "data": [
    {
      "id": "f9385g29-d21e-2391-ddab-cd8193a943ed",
      "createdAt": 1468953420000,
      "file": {
        "name": "JulieYoungUpdated.pdf",
        "ext": "pdf",
        "downloadUrl": "https://hire.lever.co/candidates/96118ae8-3d48
        "uploadedAt": 1468953420004
      },
      "parsedData": {
        "positions": [
          {
```

```
    "org": "Laugh Factory",
    "title": "VP Digital Media Productions",
    "summary": "• Went from executive assistant to the CEO to
    "location": "Los Angeles, CA",
    "start": {
      "year": 2012,
      "month": 7
    },
    "end": {
      "year": 2016,
      "month": 3
    }
  },
  {
    "org": "Intuitive Entertainment",
    "title": "Production Assistant, Production Coordinator",
    "summary": "• PA for several reality TV shows produced by
    "location": "Los Angeles, CA",
    "start": {
      "year": 2010,
      "month": 9
    },
    "end": {
      "year": 2012,
      "month": 3
    }
  },
  {
    "org": "Violet Pulications",
    "title": "Editor in Chief, Layout, Photographer",
    "summary": "• Editor in chief of college yearbook• Grew ye
    "location": "New York, NY",
    "start": {
      "year": 2008,
      "month": 9
    },
    "end": {
      "year": 2010,
      "month": 5
    }
  }
],
"schools": [
  {
    "org": "New York University Stern",
    "degree": "BS Marketing",
    "field": "Marketing, Studio Arts and Film",
    "summary": "Marketing degree with a focus on the film indu
    "start": {
      "year": 2007,
      "month": 9
    },
    "end": {
      "year": 2010,
      "month": 5
    }
  },
  {
    "org": "Savannah College of Art and Design",
    "degree": "",
    "field": "Graphic Design, Studio Arts",
    "summary": null,
    "start": {
      "year": 2006,
      "month": 9
    },
        "end": {
          "year": 2007,
          "month": 6
        }
      }
    ]
  }
},
{
  "id": "b3912e39-d22d-2958-bbad-dd2849a903dd",
  "createdAt": 1407460071043,
  "file": {
    "name": "Julie-Young.pdf",
    "ext": "pdf",
    "downloadUrl": "https://hire.lever.co/candidates/96118ae8-3d48
    "uploadedAt": 1407460071048
  },
  "parsedData": {
    "positions": [
      {
        "org": "Intuitive Entertainment",
        "title": "Production Assistant, Production Coordinator",
        "summary": "• PA for several reality TV shows produced by
        "location": "Los Angeles, CA",
        "start": {
          "year": 2010,
          "month": 9
        },
        "end": {
          "year": 2012,
          "month": 3
        }
      },
      {
        "org": "Violet Pulications",
        "title": "Editor in Chief, Layout, Photographer",
        "summary": "• Editor in chief of college yearbook• Grew ye
        "location": "New York, NY",
        "start": {
          "year": 2008,
          "month": 9
        },
        "end": {
          "year": 2010,
          "month": 5
        }
      },
      {
        "org": "Hreff Jones Publications",
        "title": "Sales & Marketing Intern",
        "summary": "• Help a new school start planning their yearb
        "location": "New York, NY",
        "start": {
          "year": 2009,
          "month": 5
        },
        "end": {
          "year": 2009,
          "month": 8
        }
      },
      {
        "org": "Little Picasso",
        "title": "Jr. Director, Summer Academy Instructor",
        "summary": "• Summer camp job working with children age 3
        "location": "Annandale, VA",
            "start": {
              "year": 2005,
              "month": 5
            },
            "end": {
              "year": 2008,
              "month": 8
            }
          }
        ],
        "schools": [
          {
            "org": "New York University Stern",
            "degree": "BS Marketing",
            "field": "Marketing, Studio Arts and Film",
            "summary": "Marketing degree with a focus on the film indu
            "start": {
              "year": 2007,
              "month": 9
            },
            "end": {
              "year": 2010,
              "month": 5
            }
          },
          {
            "org": "Savannah College of Art and Design",
            "degree": "",
            "field": "Graphic Design, Studio Arts",
            "summary": null,
            "start": {
              "year": 2006,
              "month": 9
            },
            "end": {
              "year": 2007,
              "month": 6
            }
          }
        ]
      }
    }
  ]
}
```

Download a resume file

Downloads a resume file if it exists

```
GET /candidates/:candidate/resumes/:resume/download
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/candidates/c0950a08-e88e-4435
```

Sources

A source is the way that a candidate entered into your Lever account. The most common sources in your Lever account are:

- `Posting` - Candidate applied to a posting on your careers page.
- `Referral` - Candidate was referred by an employee at your company.
- `Add New` - Candidate was added manually into your Lever account in the app.
- `Email Applicant` - Candidate was added via applicant@hire.lever.co email address.
- `Email Lead` - Candidate was added via lead@hire.lever.co email address.
- `LinkedIn` - Candidate was added from LinkedIn using the Lever Chrome Extension.
- `GitHub` - Candidate was added from GitHub using the Lever Chrome Extension.
- `AngelList` - Candidate was added from AngelList using the Lever Chrome Extension.

Attributes

text	`Gild` String Source text
count	`24` Integer Number of candidates tagged with this source

List all sources

```
GET /sources
```

Lists all sources in your Lever account.

Examples

```
curl -u API_KEY: https://api.lever.co/v1/sources
```

```
{
  "data": [
    {
      "text": "Gild",
      "count": 24
    },
    {
      "text": "Posting",
      "count": 51
    },
    {
      "text": "Referral",
      "count": 90
    },
    {
      "text": "Email Applicant",
      "count": 135
    },
    {
      "text": "Email Lead",
      "count": 83
    }
  ]
}
```

Stages

Stages are steps in the recruiting workflow of your hiring pipeline. All candidates belong to a stage and change stages as they progress through the recruiting pipeline, typically in a linear fashion.

Attributes

id `7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5`
String
Stage UID

text `New applicant`
String
Title of the stage

Retrieve a single stage

```
GET /stages/:stage
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/stages/7623c5f3-9a18-4ccf-a24
```

```
{
  "data": {
    "id": "7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5",
    "text": "New applicant"
  }
}
```

List all stages

Lists all pipeline stages in your Lever account.

```
GET /stages
```

Examples

```
curl -u API_KEY: https://api.lever.co/v1/stages
```

```
{
  "data": [
    {
      "id": "7623c5f3-9a18-4ccf-a24f-a7c1ad4163f5",
      "text": "New applicant"
    },
    {
      "id": "129db616-7430-434f-adfa-75ecb75d1870",
      "text": "New lead"
    },
    {
      "id": "32f3a65a-899a-4fc3-b2a7-1dbe8300b1a4",
      "text": "Recruiter Screen"
    },
    {
      "id": "c449d622-9e1a-4a59-bfba-546eb4181e08",
      "text": "Phone Interview"
    },
    {
      "id": "64091991-e4ea-446c-b9c2-6bbb07457985",
      "text": "On-Site Interview"
    },
    {
      "id": "c7b32d79-005f-4941-bf7c-d52d687affa1",
      "text": "Background Check"
    },
    {
      "id": "62e2491f-e5e1-4c3d-b8d0-8886e7282285",
      "text": "Offer"
    }
  ]
}
```

Tags

Tags provide additional information or context to a candidate within your pipeline. Tags serve as a way of grouping candidates for easy viewing of individuals with specific attributes.

Attributes

text	`Infrastructure Engineer` String Tag text
count	`23` Integer Number of candidates tagged

List all tags

```
GET /tags
```

Lists all tags in your Lever account.

Examples

```
curl -u API_KEY: https://api.lever.co/v1/tags
```

```
{
  "data": [
    {
      "text": "Infrastructure Engineer",
      "count": 23
    },
    {
      "text": "Customer Success Manager",
      "count": 15
    },
    {
      "text": "Customer Success",
      "count": 31
    },
    {
      "text": "Full-time",
      "count": 66
    },
    {
      "text": "San Francisco",
      "count": 70
    }
  ]
}
```

Users

Users in Lever include anyone who has been invited to join in on recruiting efforts. There are five different access roles in Lever. From most to least access, these roles are Super admin, Admin, Team member, Limited team member, and Interviewer. For more specific information about access roles in Lever, check out this article.

Attributes

id `d2b18d51-d4a0-4976-a614-9e481577892a`
String
User UID

name `Josh Ackerman`
String
User's preferred name

username `josh`
String
Username, extracted from user's email address

email `josh@example.com`
String
User's email address

createdAt `1407357447018`

Timestamp [?]

Datetime when user was created

deactivatedAt

`1409556487918`

Timestamp [?]

Datetime when user was deactivated, `null` for an active user

accessRole

`super admin`

string

User's access role. One of: 'super admin', 'admin', 'team member', 'limited team member', 'interviewer' [?]

photo

`https://gravatar.com/avatar/ff781412e2bb44142addf4258 s=26&d=404`

string

URL for user's gravatar, if enabled [?]

externalDirectoryId

`2277399`

String

optional

Unique Id for user in external HR directory

Retrieve a single user

This method returns the full user record for a single user.

```
GET /users/:user
```

Examples

Retrieve a user by user ID

```
curl -u API_KEY: https://api.lever.co/v1/users/d2b18d51-d4a0-4976-a614
```

```
{
  "data": {
    "id": "d2b18d51-d4a0-4976-a614-9e481577892a",
    "name": "Josh Ackerman",
    "username": "josh",
    "email": "josh@example.com",
    "createdAt": 1407357447018,
    "deactivatedAt": 1409556487918,
    "externalDirectoryId": "2277399",
    "accessRole": "super admin",
    "photo": "https://gravatar.com/avatar/ff781412e2bb44142addf42589a0
  }
}
```

List all users

Lists all the users in your Lever account. Only active users are included by default.

```
GET /users
```

Parameters

email

`josh@example.com`

String

optional

Filter results to users that match an email address. Provided email must exactly match the canonicalized version of the user's email.

includeDeactivated

`true`

Boolean

optional

Include deactivated users along with activated users.

Examples

```
curl -u API_KEY: https://api.lever.co/v1/users
```

```
{
  "data": [
    {
      "id": "d2b18d51-d4a0-4976-a614-9e481577892a",
      "name": "Josh Ackerman",
      "username": "josh",
      "email": "josh@example.com",
      "createdAt": 1407357447018,
      "deactivatedAt": 1409556487918,
      "externalDirectoryId": "2277399",
      "accessRole": "super admin",
      "photo": "https://gravatar.com/avatar/ff781412e2bb44142addf42589
    },
    {
      "id": "133a52cc-6f8a-4b33-bfad-d7bc92e1ea23",
      "name": "Rachael Stedman",
      "username": "rachael",
      "email": "rachael@example.com",
      "createdAt": 1478035107621,
      "deactivatedAt": null,
      "externalDirectoryId": "2277380",
      "accessRole": "super admin",
      "photo": "https://gravatar.com/avatar/ff781412e2bb44142addf42589
    }
  ]
}
```

Find users that match a provided email address

```
curl -u API_KEY: https://api.lever.co/v1/users?email=josh%40example.co
```

```
{
  "data": [
    {
      "id": "d2b18d51-d4a0-4976-a614-9e481577892a",
      "name": "Josh Ackerman",
      "username": "josh",
      "email": "josh@example.com",
      "createdAt": 1407357447018,
      "deactivatedAt": 1409556487918,
      "externalDirectoryId": "2277399",
      "accessRole": "super admin",
      "photo": "https://gravatar.com/avatar/ff781412e2bb44142addf42589
    }
  ]
}
```

Create a user

```
POST /users
```

This endpoint enables integrations to create users in your Lever account.

Users will be created with the Interviewer access role by default. Users may be created with Interviewer, Limited Team Member, Team Member, Admin, or Super Admin access.

Lever accepts requests of type `application/json`.

Note: This will not send an invite to the user, so direct auth users will need to go through the direct auth password flow.

Parameters

name
`Josh Ackerman`
String
required
User's preferred name

email
`josh@example.com`
String
required
User's email address

accessRole
`team member`
String
optional
User's access role. One of: 'super admin, 'admin', 'team member', 'limited team member', 'interviewer' [?]

externalDirectoryId
`2277399`
String
optional
Unique Id for user in external HR directory

Examples

```
curl -H "Content-Type: application/json" -X POST -u API_KEY: -d '{
  "name": "Josh Ackerman",
  "email": "josh@example.com",
  "accessRole": "super admin",
  "externalDirectoryId": "2277399"
}' "https://api.lever.co/v1/users"
```

```
201 - Created

{
  "data": {
    "id": "d2b18d51-d4a0-4976-a614-9e481577892a",
    "name": "Josh Ackerman",
    "username": "josh",
    "email": "josh@example.com",
    "createdAt": 1407357447018,
    "deactivatedAt": 1409556487918,
    "externalDirectoryId": "2277399",
    "accessRole": "super admin",
    "photo": "https://gravatar.com/avatar/ff781412e2bb44142addf42589a0
  }
}
```

Update a user

When you PUT data, Lever expects you to send the entire resource. Every field will be overwritten by the body of the PUT request. If you don't include a field, it will be deleted or reset to its default. Be sure to include all fields you still want to be populated. name, email, and accessRole are required fields. Note that resetting accessRole to `interviewer` will result in a user losing all of their followed profiles.

```
PUT /users/:user
```

Examples

```
curl -H "Content-Type: application/json" -X POST -u API_KEY: -d '{
  "name": "Josh Ackerman",
  "email": "josh@example.com",
  "accessRole": "super admin",
  "externalDirectoryId": "2277399"
}' "https://api.lever.co/v1/users/1ee63c8a-f15a-4c76-83b5-a3b37dca775f
```

```
201 - Created

{
  "data": {
    "id": "d2b18d51-d4a0-4976-a614-9e481577892a",
    "name": "Josh Ackerman",
    "username": "josh",
    "email": "josh@example.com",
    "createdAt": 1407357447018,
    "deactivatedAt": 1409556487918,
    "externalDirectoryId": "2277399",
    "accessRole": "super admin",
    "photo": "https://gravatar.com/avatar/ff781412e2bb44142addf42589a0
  }
}
```

Deactivate a user

Deactivated users remain in the system for historical record keeping, but can no longer log in and use Lever.

```
POST /users/:user/deactivate
```

Examples

```
curl -H "Content-Type: application/json" -X POST -u API_KEY: https://a
```

```
200 - OK

{
  "data": {
    "id": "d2b18d51-d4a0-4976-a614-9e481577892a",
    "name": "Josh Ackerman",
    "username": "josh",
    "email": "josh@example.com",
    "createdAt": 1407357447018,
    "deactivatedAt": 1409556487918,
    "externalDirectoryId": "2277399",
    "accessRole": "super admin",
    "photo": "https://gravatar.com/avatar/ff781412e2bb44142addf42589a0
  }
}
```