

×

COLLAPSE ALL

GETTING STARTED

Introduction

Authentication with API keys

Authentication with OAuth

Specifying Filter Parameters

Timezone Offsets

Fields

Updating Specific Fields

Pagination

Webhooks

Rate Limits

HTTP Response Codes

API Clients

RESOURCES

Advanced Filtering

Leads

Contacts

Activities

Call

Created

Email

EmailThread

LeadStatusChange

Meeting

Note

OpportunityStatusChange

SMS

TaskCompleted

Opportunities

Tasks

Memberships

Users

Organizations

Roles

Custom Fields

Lead Custom Fields

Contact Custom Fields

Opportunity Custom Fields

Activity Custom Fields

Shared Custom Fields

Custom Field Schemas

Lead Statuses

Opportunity Statuses

Pipelines

Groups

Reporting

Email Templates

SMS Templates

Connected Accounts

Send As

Sequences

Dialer

Smart Views

Bulk Actions

Integration Links

Exports

Phone Numbers

Event Log

List of Events

Retrieve Events

Webhook Subscriptions

Webhook Filters

Scheduling Links

User Scheduling Links

Shared Scheduling Links

Scheduling Link Associations

Custom Activities

Custom Activity Types

Custom Activity Instances

CHANGELOG

API Changelog

Product Changelog

QUICK LINKS

Log in to Close

Visit our Homepage

Close Support Documentation

The Making of Close

Leads

Leads are the most important object in Close. They represent a company or organization and can contain contacts, tasks, opportunities, and activities. In fact, these other objects *must* be children of a Lead. You can think of a Lead in Close like both a "lead" and "account" in the terminology of some other CRMs.

When a lead is returned, its basic info as well as related tasks, opportunities, and custom fields are included. Activities are excluded and have to be fetched separately via the [activities](#) endpoint.

Custom fields that have a value are returned as `custom.FIELD_ID`, where FIELD_ID is the ID of the custom field (see [Custom Fields](#) for more details). Note that using the `custom` field dict is deprecated, and the field will be removed from the API.

When using the `_fields` parameter, specify `custom` to show all custom fields. Returning individual custom fields is currently not supported via `_fields`.

[Smart Fields](#) are not included in the response by default. If you'd like to include them, you can:

- Fetch a specific Smart Field with `?_fields=smart_field_name,other_lead_field_names`
- Fetch all Lead fields and all Smart Fields with `?_fields=_all`

To speed up your integration, we recommend that you only use the `_all` flag to discover what Smart Fields are available and then use `?_fields=specific,set,of,fields` to return only the fields you need.

Filtering leads

To easily find Leads that match specific conditions, use the [Advanced Filtering API](#).

List Leads

GET /lead/{?_limit, _skip, _fields}

Example Request/Response

Create a new lead.

POST /lead/

Contacts, addresses, and custom fields can all be nested in the lead. Currently, activities, tasks, and opportunities must be posted separately.

status and *status_id* (optional)

- Optionally post either `status` or `status_id` (but not both). If neither property exists, the organization's default (first) status will be used when creating new leads.
- (Recommended) Use `status_id` (like in the example payload), so that users can rename statuses in the UI without breaking your implementation.
- Alternatively, you can post a `status` containing the label of an already-created status object.

custom.FIELD_ID (optional)

- You can set custom fields by setting `custom.FIELD_ID` to the field value, where FIELD_ID is the ID of the custom field, e.g.:

```
1 {
2   "custom.cf_v6S011I6MqcbVvB2FA5Nk8dr5MkL8sWuCIG8cUle09c": "value",
3   "custom.cf_8wtBWsdRU2Fur7GDnEeXQ7ra2Vu7R4hG1SNYdiEhh0F": "other value"
4 }
```

- Note that if a custom field has `accepts_multiple_values: true`, the entire value will be replaced. For example, given a Lead has a Custom Choice Field with value `["A", "B"]`, adding choice `"C"` would mean setting the value to `["A", "B", "C"]`.
- Note that using the `custom` field dict or the `custom.FIELD_NAME` syntax (instead of custom field IDs) is deprecated and will be removed from the API.
- See the [Custom Fields](#) paragraph for more details.

Example Request/Response

Retrieve a single lead.

GET /lead/{id}/{?_fields}

Example Request/Response

Update an existing lead.

PUT /lead/{id}/

Supports non-destructive patches.

status:

- See description for `status` and `status_id` in `POST /lead/` above.
- custom.FIELD_ID* (optional)
- See description for `custom.FIELD_ID` in `POST /lead/` above.
- Additionally, you can unset a single field by using `{ "custom.FIELD_ID": null }`, e.g.:

```
1 { "custom.cf_v6S011I6MqcbVvB2FA5Nk8dr5MkL8sWuCIG8cUle09c": null }
```

- If the custom field you are trying to update accepts multiple values, you can also specify `.add` or `.remove` as part of the field key to add/remove a single value to/from a list of values, e.g.:

```
1 { "custom.cf_v6S011I6MqcbVvB2FA5Nk8dr5MkL8sWuCIG8cUle09c.add": "Wednesday" }
```

adds "Wednesday" to the list of values which already exist on the given lead.

Example Request/Response

Delete a lead.

DELETE /lead/{id}/

Example Request/Response

Merge two leads.

POST /lead/merge/

For details and definitions of 'source' and 'destination', see the Merge Leads feature in the UI.

Example Request/Response