

×

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 

Tasks

Tasks are action items with a given date that are assigned to a sales rep.

Incomplete tasks (`is_complete` is false) show in the sales rep's inbox, whereas complete tasks (`is_complete` is true) are shown in the archive. Archived tasks of certain types are automatically deleted after a certain amount of time.

The `date` of a task represents the point in time of when the task is actionable and appears in the inbox. It can either be date-only (e.g. `2015-01-05`), or a date time (e.g. `2015-01-10T05:00:00+00:00`). When ordering tasks by date, date-only tasks are ordered before any date-time tasks at the given date, taking the time zone (`x-tz-offset`) into account. For example, in Pacific Standard Time (`x-tz-offset: -8`) and ascending order, a task dated `2015-01-12` will appear before `2015-01-12T08:00:00`, but after `2015-01-12T07:00:00`. Some endpoints accept and return a `due_date` field, which is deprecated and shouldn't be used.

The type of a task is returned using the `_type` field, which can also be used to filter tasks. When not filtering by `_type`, only tasks of `lead` type are returned. All tasks can be shown using `_type=all`, and multiple task types can be filtered by passing the type names comma separated into `_type__in` (e.g. `_type__in=missed_call,voicemail`). Some task types have a corresponding object which is returned using the `object_id` and `object_type` params. The following is a list of task types and corresponding objects that exist:

- `lead`: A to-do item/task for a given lead. Both `object_type` and `object_id` are null, since the lead is already returned in `lead_id`.
- `incoming_email`: One or multiple incoming emails on an email thread. The `object_type` is `emailthread`, and `object_id` is the ID of the corresponding email thread. Multiple unread emails in one thread are consolidated into one task. The `emails` array contains a list of email activity IDs related to this task. The `subject` field contains the subject of the email thread.
- `email_followup`: A reminder to follow up on a specific email you sent in the past and didn't get a response to. The `object_type` is `emailthread`, and `object_id` is the ID of the corresponding email thread. A reference to the email you sent originally is stored in `email_id`. The `subject` and `body_preview` fields contain data related the email you should follow up on.
- `missed_call`: A missed call. The `object_type` is `call`, and `object_id` is the activity ID of the call that was missed. The `phone` field contains the phone number of the remote party, the `local_phone` field contains the number that was called.
- `answered_detached_call`: A call from a number that isn't associated with any of the organization's contacts. The `object_type` is `call`, and `object_id` is the ID of the corresponding call. The `phone` field contains the phone number of the remote party.
- `voicemail`: A voicemail. Just like `missed_call`, with the additional fields `voicemail_duration` and `voicemail_url`.
- `opportunity_due`: Indicates that an opportunity is scheduled to close on this date. The `object_type` is `opportunity`, and `object_id` is the ID of the corresponding opportunity.
- `incoming_sms`: An incoming SMS. The `object_type` is `sms`, and `object_id` is the ID of the corresponding SMS. The `remote_phone` field contains the phone number of the sender, the `local_phone` field contains the number that was texted. If incoming SMS is actually an MMS, it will have all the attachments in the `attachments` field: check the example response for what fields could be present there.

List or filter tasks.

```
GET /task/{?id, id__in, _type, lead_id, is_complete, date__[<|>|lte|gte]},
date_created__[<|>|lte|gte], assigned_to, view, _order_by}
```

The `view` parameter can be used to conveniently access the different task views. It accepts one of the following values:

- `inbox`: Shows incomplete tasks up to the end of the user's day only (taking the time zone into account).
- `future`: Shows incomplete tasks starting from the user's tomorrow only (taking the time zone into account).
- `archive`: Shows complete tasks only.

The `_order_by` field allows ordering by `date` or `date_created` (descending order by prepending a minus, e.g. `_order_by=-date`).

▼

Example Request/Response

Create a task.

```
POST /task/
```

Currently only tasks of type `lead` can be created, so the `_type` field may be omitted.

▼

Example Request/Response

Bulk-update tasks.

```
PUT /task/{?...}
```

Any of the filters of the GET endpoint may be used. For example, to update multiple tasks with given IDs A, B and C, `id__in=A,B,C` would be passed.

Only the `assigned_to`, `date` and `is_complete` fields may be updated.

▼

Example Request/Response

Fetch a task's details.

```
GET /task/{id}/
```

▼

Example Request/Response

Update a task.

```
PUT /task/{id}/
```

The `assigned_to`, `date` (either a date or a date-time) and `is_complete` fields may be updated on all tasks. If the task type is `lead`, the `text` field may also be modified.

▼

Example Request/Response

Delete a task.

```
DELETE /task/{id}/
```

▼

Example Request/Response