Close DOCS

Close API Documentation

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

Rate Limits

We enforce API call rate limits to protect our infrastructure from excessive request rates, to keep Close fast and stable for everyone. These limits are high enough that typical API workflows aren't affected. However, please do code your integration to follow the rule below:

If you receive a response status code of 429 (Too Many Requests), please sleep/pause for the number of seconds specified by the `rate_reset` value before making additional requests to that endpoint. See below for the specific response format.

Rate limits are enforced per endpoint group. Endpoint groups are used to provide more granular control by grouping endpoint URL paths and methods (e.g. GET, PUT, etc.) together. For instance, GETs to /api/v1/lead/ and POSTs/PUTs to /api/v1/activity/ may be counted as two different API groups. This allows us to offer a higher limit on lightweight requests than we would be able to on more resource intensive request types.

API requests are limited at a per Organization level across all users' API keys. We also enforce a lower rate limit per API key, which helps ensure that an individual heavy integration doesn't cause other lightweight integrations (on separate API keys) to get rate limited unnecessarily. The per Organization limit is currently 3 times higher than individual API key rate limits, meaning that if the API key rate limit maximum requests per second (RPS) is 20 RPS, the organization wide limit would be 60 RPS for that same endpoint group. This allows you to use 3 API keys at their maximum RPS and not hit the Organization rate limit until you add a 4th key making additional requests. The 429 response provides details to identify which limit and endpoint group was hit.

Most API responses will have the following headers to provide rate limiting statistics about the limit it's closest to hitting.

- `x-rate-limit-limit`: Request limit enforced for this endpoint, some endpoints may allow bursting over this limit
- `x-rate-limit-remaining`: Requests left in the enforcement window
- `x-rate-limit-reset`: Seconds remaining before this enforcement window ends (as a decimal).

Each `429` response is guaranteed to have the `x-rate-limit-reset` header set as well as the `retry-after` header as per RFC 7231 (equivalent to `x-rate-limit-reset` rounded up to the next integer), and provides the following data in its body:

Copy

```
{
  "error": {
    "message": "API call count exceeded for this period",
    "rate_reset": 0.870663,     # Seconds remaining until the next enforcement window starts

    # The fields below may not always be set.
    "rate_limit": 40,           # Request limit enforced for this endpoint and rate limit type (key or org)
    "rate_window": 1,           # Number of second(s) in enforcement window
    "rate_limit_type": "key",   # Type of rate limit hit ("key" or "org")
    "rate_endpoint_group": "99ad0b85407fbfce6882152c4cd0b86d"  # Endpoint group ID
  }
}
```

Some API endpoints may have stricter (unpredictable) rate limits and may trigger a 429 even if the user agent places requests within the enforcement window. In these scenarios only the `rate_reset` values and `retry-after` header may be set.

Note: We recommend using the `rate_reset` value instead of the `retry-after` header for a more accurate wait time.

PREVIOUS
Webhooks

NEXT
HTTP Response Codes