HIVER API

Search

Hiver API

If you need any help, you can refer to our help article here.

Overview

To get started you can get the API keys created by referring to the `Getting API Key` section below. You need these API keys to access any of the APIs listed below.

Getting Started

Getting API key

You need to create an API key to access the Hiver APIs. Please follow these steps to get an API key

- Go to Developer settings in the left panel in the Hiver admin panel.
- Click on Create app.
- Enter a name for the app and click on Create to Generate the API keys.
- You’ll see the app created with the API key. Click on the copy icon to get the API key copied. These API keys can be used as a bearer token as follows

```
curl 'https://api2.hiverhq.com/v1/inboxes'  --header 'Authorization: Bearer <your-access-token>
```

If you suspect that any of your API keys have been compromised, you must take immediate action and delete them from the Hiver admin panel.

Authentication

You need to have access to a Hiver API key to access any of these Hiver APIs. Do note that the API key can be created by an administrator in Hiver. The API key will have the access privileges of an admin user. It is very important to store and use your API keys safely.

Request Format

Hiver uses HTTP methods like `GET`, `POST`, `PATCH` etc for different types of operations. Depending on the type of the request, the query parameters, request body, URL path needs to be formed.
For more details, refer to individual APIs in the API reference section below.

Response format

The response will be a JSON object. The usual object be in the format as follows, enclosed in a data object

```
    "data":{
      ...
    }
  }
```

The response for list will be as follows:

```
  {
    "results": [
      {..},
      ],
    "pagination": {
      "next_page": "<next-page-token>"
    }
  }
```

For the paginated results, use the next page token in the subsequent request to get further results. Keep continuing to get the next page until the next_page returns `null`

Pagination

Some endpoints return a list like list of inboxes etc. These endpoints return paginated results. To get the complete list, you need to implement the fetching of paginated lists.
These endpoints support following parameters

Query Parameter	Description
limit	Limits the number of results in a page. Supported values are 10-100. Defaults to 10
sort_by	The field based on which the result need to be sorted. Refer to individual APIs to find the supported fields
sort_order	Ascending, Descending order. Supported values: `asc`, `desc`. Defaults to `asc`

All of these field are optional, depending on the endpoint being used.

Errors

The error response will have an appropriate HTTP status code attached in the response. In addition to this, there could also be message, in some cases, to help identify the errors

Status Code	Description
200	Success
204	Success - No response
400	Bad request - Request sent without required params
401, 403	Forbidden - Not authorised to access
404	Not found - Request entity not found
429	Too many requests
500	Internal server error - If this happens, get in touch with us, or watch for any degradation in status.hiverhq.com

The error messages will be in one of the following formats:

```
{
  "errors": [
    {"message": "Actual Error message"}
    ..
  ]
}
or
```

```
}
```

Refer to the rate-limiting section to get to know more about rate-limiting enforced by Hiver.

Rate limits

Hiver enforces rate-limits based per account. The current rate is limited at 1 RPS (request per second). Once you hit the limit, you'll start getting `429` `Too Many Requests` error in the response. If you get this error, retry with an exponential backoff. Continuous retries with 429 error could result in the client-ip/api-key getting blacklisted. Also, the APIs are limited to 5000 requests per day. Please get in touch with us if you want to increase this.

BASE URL

Production
https://api2.hiverhq.com/v1

LANGUAGE BOX

cURL Ruby Python PHP Java Node.js Go .NET

Was this section helpful? Yes No

Inbox

An Inbox is an entity that has conversations & users. Most common form of an Inbox is a shared mailbox (https://hiverhq.com/shared-inbox) which manages emails. The APIs only manage the shared mailbox. We plan to add other type of inboxes in the APIs soon. It can have tags to help manage the conversations better.
Following is an example of an Inbox object

```
    "id": "101",
    "display_name": "Customer Support",
    "channel_type": "email",
    "email": "info@hiver.com",
    "inbox_type": "user",
    "is_authorised": false,
    "source_user": {
      "id": "765676",
      "email": "info@hiver.com"
    },
    "created_at": 176878888,
    "updated_at": 176878888
  }
```

Following is an example of an Inbox User object

```
  {
    "id": "456",
    "first_name": "Phoebe",
    "last_name": "Buffay",
    "email": "p.buffay@friends.com",
    "phone_number": "+19876543444",
    "is_joined": false
  }
```

Following is an example of an Inbox Tag object

```
  {
    "id": "56789",
    "name": "Priority",
    "color_code": "#ce93d8",
    "type": "user",
    "created_at": 1708945347
  }
```

List all the inboxes

HEADER PARAMETERS

Authorization string

RESPONSES

Expand all

200 Object

successful operation

RESPONSE ATTRIBUTES

results array

+ Show child attributes

pagination object

+ Show child attributes

GET /inboxes

cURL

```
curl --location 'https://api2.hiverhq.com/v1/inboxes' \
--header 'Authorization: Bearer {token}'
```

Try in API Explorer

RESPONSE

200

```
{
  "results": [
    {
      "id": "101",
        "inbox_type": "user",
        "is_authorised": false,
        "source_user": {
```

Show more

Was this section helpful? Yes No

Get an inbox by id

Get an Inbox by Id

HEADER PARAMETERS

Authorization string

PATH PARAMETERS

inbox_id integer (int64) REQUIRED

ID of the inbox to get details

RESPONSES

Expand all

200 Object

successful operation

RESPONSE ATTRIBUTES

data object

+ Show child attributes

Was this section helpful? Yes No

GET /inboxes/{inbox_id} cURL

```
curl --location --globoff 'https://api2.hiverhq.com/v1/inboxes/{inbox_id}' \
--header 'Authorization: Bearer {token}'
```

Try in API Explorer

RESPONSE

200

```
{
  "data": {
    "id": "101",
    "display_name": "Customer Support",
    "channel_type": "email",
    "email": "info@example.com",
    "inbox_type": "user",
    "is_authorised": false,
    "source_user": {
      "id": "12323",
      "email": "source@example.com"
```

Show more

Get all users in the inbox

HEADER PARAMETERS

Authorization string

PATH PARAMETERS

inbox_id string REQUIRED

ID of inbox to get the user list for

RESPONSES

Expand all

200 Object

Successful Operation

RESPONSE ATTRIBUTES

results array

+ Show child attributes

pagination object

+ Show child attributes

GET /inboxes/{inbox_id}/users cURL

```
curl --location 'https://api2.hiverhq.com/v1/inboxes/343/users' \
--header 'Authorization: Bearer {token}'
```

Try in API Explorer

200

```
{
  "results": [
    {
      "id": "456342",
      "first_name": "Phoebe",
      "last_name": "Buffay",
      "email": "p.buffay@friends.com",
      "phone_number": "+19876543444",
      "is_joined": false
    }
  ],
```

Was this section helpful? Yes No

Search users in the inbox

QUERY PARAMETERS

email string REQUIRED

Email address to be searched for by exact match

PATH PARAMETERS

inbox_id string REQUIRED

ID of inbox to get the user list for

RESPONSES

Expand all

200 Object

Successful Operation

RESPONSE ATTRIBUTES

results array

+ Show child attributes

pagination object

+ Show child attributes

GET /inboxes/{inbox_id}/users/search?email=bob@example.com cURL

```
curl --location 'https://api2.hiverhq.com/v1/inboxes/343/users/search?email=bob@example.com'
```

Try in API Explorer

200

```
{
  "results": [
    {
      "id": "456342",
      "first_name": "Phoebe",
      "last_name": "Buffay",
      "email": "p.buffay@friends.com",
      "phone_number": "+19876543444",
      "is_joined": false
    }
  ],
```

Was this section helpful? Yes No

Get tags in the inbox

PATH PARAMETERS

inbox_id string REQUIRED

ID of inbox to get the user list for

RESPONSES

Expand all

200 Object

Successful Operation

RESPONSE ATTRIBUTES

results array

+ Show child attributes

pagination object

+ Show child attributes

GET /inboxes/{inbox_id}/tags cURL

```
curl --location 'https://api2.hiverhq.com/v1/inboxes/343/tags'
```

Try in API Explorer

RESPONSE

200

```
{
  "results": [
    {
      "id": "56789",
        "created_at": 1708945347
      }
    ],
    "pagination": {
```

Was this section helpful? Yes No

Search tags in the inbox

QUERY PARAMETERS

name string REQUIRED

Name of the tag to be searched for

PATH PARAMETERS

inbox_id string REQUIRED

ID of inbox to get the user list for

RESPONSES

Expand all

200 Object

Successful Operation

RESPONSE ATTRIBUTES

results array

+ Show child attributes

pagination object

+ Show child attributes

GET /inboxes/{inbox_id}/tags/search?name=Priority cURL

```
curl --location 'https://api2.hiverhq.com/v1/inboxes/343/tags/search?name=Priority'
```

Try in API Explorer

200

```
{
  "results": [
    {
      "id": "56789",
      "name": "Priority",
      "color_code": "#ce93d8",
      "type": "user",
      "created_at": 1708945347
    }
  ],
  "pagination": {
```

Was this section helpful? Yes No

Conversations

A conversation can be part of one or more inboxes. It can be assigned to a member of the inbox, have a status and tags associated with it.

Following is an example of a Conversation object

Was this section helpful? Yes No

Show more