



# Confluent Cloud APIs

Search...

Download OpenAPI specification: [Download](#)

## Introduction

Confluent Cloud: [support@confluent.io](mailto:support@confluent.io)

Object Model: <https://www.confluent.io/cloud-contact-us/>

## Authentication

[Introduction](#) >

## Errors

## Page

### Note

This documents the collection of Confluent Cloud APIs. Each API documents its [lifecycle phase](#). APIs marked as Early Access or Preview are not ready for production usage. We're currently working with a select group of customers to get feedback and iterate on these APIs.

## Data

Confluent Cloud APIs are a core building block of Confluent Cloud. You can use the APIs to manage your own account or to integrate Confluent into your

## HTTP Guidelines

[product](#) >

Most of the APIs are organized around [REST](#) and the resources which make up Confluent Cloud. The APIs have predictable resource-oriented URLs, transport data using JSON, and use standard HTTP verbs, response codes, authentication, and design principles.

## IDENTITY ACCESS MANAGEMENT (V2)

## Object Model

### The API Keys Model

### Note

This section describes the object model for many Confluent Cloud APIs, but not all. The Connect v1 API group has a different object model. You can review the example request and response bodies in [Connect v1 API](#) to see its object model.

## GET Read an API Key

Confluent Cloud APIs are primarily designed to be declarative and intent-oriented. In the `PUT` and `POST` keywords, tell the API what you want (for example, throughput or SLOs) and it will figure out how to make it happen (for example, cluster sizing).

Confluent objects act as a "record of intent" — after you create the object, Confluent Cloud will work tirelessly in the background to ensure that the object exists as specified.

Confluent APIs represent objects in JSON with media-type `application/json`.  
Service Accounts (iam/v2) >

Many objects follow a model consisting of `spec` and `status`. An object's `spec` tells Confluent the *desired state* (specification) of the resource. The object may not be immediately available or changes may not be immediately applied. For this reason, many objects also have a `status` property that provides info about the *current state* of the resource. Confluent Cloud is continuously and actively managing each resource's current state to match its desired state.

Role-Based Access Control (rbac/v2) >

- `api_version` — API objects have an `api_version` field indicating their API version.
- `kind` — API objects have a `kind` field indicating the kind of object it is.
- `id` — Each object in the API will have an identifier, indicated via its `id` field, and should be treated as an opaque string unless otherwise specified.

There are a number of other [standard properties](#) and that you'll encounter used by many API objects. And of course, objects have plenty of non-standard fields that are specific to each object *kind*... this is what makes them interesting!

## Authentication

[Certificate Authorities](#) >

Confluent uses API keys and JSON Web Tokens (JWTs) to integrate your applications and workflows to your Confluent Cloud resources using the Confluent Cloud REST APIs. Your applications and workflows must be authenticated and authorized in order to access and manage Confluent Cloud resources.

## API keys

You can create and manage your API keys using the Confluent Cloud Console or Confluent CLI. For more information, see [Use API Keys to Control Access in Confluent Cloud](#).

Confluent Cloud uses the following two categories of API keys:

- A Cloud API key grants access to the Confluent Cloud Management APIs, such as for Provisioning and Metrics integrations.
- A resource-specific API key grants access to a Confluent Kafka cluster (Kafka API key), a Confluent Cloud Schema Registry (Schema Registry API key), Flink (Flink API key scoped to an Environment + Region pair), or a ksqldb application.

Each Confluent Cloud API key is associated with a principal (specific user or service account) and inherits the permissions granted to the owner.

- For example, if service account `Armageddon` is granted ACLs on Kafka cluster `neptune`, then a Kafka API Key for `neptune` owned by `Armageddon` will have these ACLs enforced.
- Note: API keys are automatically deleted when the associated user or service account is deleted (for example, when an employee leaves the company or moves to a new department and an SSO integration removes the Confluent Cloud user as they no longer require access).
- Confluent strongly recommends that you use service accounts for all production-critical access.

Confluent Cloud API keys grant access to Confluent Cloud resources, so keep them secure! Do not share your API keys and secrets in publicly-accessible locations, such as GitHub or client-side code.

All API requests must be made over HTTPS. Calls made over plain HTTP will fail. API requests without authentication will also fail.

To use an API key, you must send it in an `Authorization: Basic {credentials}` header. Remember that HTTP Basic authentication requires you to provide your credentials as the API key ID and associated API secret separated by a colon and encoded using Base64 format. For example, if your API key ID is

`ABCDEFGHI123456789` and the API key Secret is `XNCIW93I2LISQFJSJ823K1LS902KLDFMCZPWEQ`, then the authorization header is:

```
Authorization: Basic QUJDREVGR0gxMjMONTY3ODk6WE5DSVc5M0kyTDFTUVBKU0c4MjNlMUx
```

You can generate this header example from the API key:

macOS:

```
$ echo -n "ABCDEFGHI123456789:XNCIW93I2LISQFJSJ823K1LS902KLDFMCZPWEQ" | base64
```

Linux:

```
$ echo -n "ABCDEFGHI123456789:XNCIW93I2LISQFJSJ823K1LS902KLDFMCZPWEQ" | base64
```

To find out if an API operation supports Cloud API Keys, look in the AUTHORIZATIONS listing for `cloud-api-key`.

To find out if an API operation supports resource-specific API Keys, look in the AUTHORIZATIONS listing for `resource-api-key`.

## External OAuth

You can use [OAuth/OIDC support for Confluent Cloud](#) to authenticate and authorize access to applications and workloads for the following Confluent Cloud REST APIs:

- Kafka REST API: [Kafka REST API for Clusters\(V3\)](#). For an API overview and examples, see [Cluster Management with Kafka REST API](#).
- Schema Registry REST API: [Schema Registry REST API for Schemas\(V1\)](#) and [Subjects](#). For an API overview and examples, see [Schema Registry REST API for Confluent Cloud](#).

Alternatively, to find out if an API operation supports external tokens, look in the AUTHORIZATIONS listing for `external-access-token`.

## Confluent STS tokens

Confluent Security Token Service (STS) issues access tokens ( `confluent-sts-access-token` ) by exchanging an external token ( `external-access-token` ) for a `confluent-sts-access-token`. You can use Confluent STS tokens to authenticate to Confluent Cloud APIs that support the `confluent-sts-access-token` notation.

To find out if an API operation supports Confluent STS tokens, look in the AUTHORIZATIONS listing for `confluent-sts-access-token`.

## Partner OAuth

Approved partners can fetch Partner tokens ( `confluent-partner-access-token` ) that validate their identity and grant access to the Partner API ( `partner/v2` ), which lets them sign up an organization on behalf of a customer, manage entitlements (create, read, and list), and read or list organizations they have signed up.

To find out an API operation supports Partner tokens, look in the AUTHORIZATIONS listing for `confluent-partner-access-token`.

## cloud-api-key

Authenticate with Cloud API Keys using HTTP Basic Auth. Treat the Cloud API Key ID as the username and Cloud API Key Secret as the password.

**Security Scheme Type:** HTTP

**HTTP Authorization Scheme:** `basic`

---

## confluent-sts-access-token

Authenticate with Confluent API using this credentials (JSON Web Tokens) following OAuth 2.0.

**Security Scheme Type:** OAuth2

**Flow type:** `clientCredentials`

**Token URL:** `https://api.confluent.cloud/sts/v1/oauth2/token`

---

## api-key

Authenticate with API Keys using HTTP Basic Auth. Treat the API Key ID as the username and API Key Secret as the password.

**Security Scheme Type:** HTTP

**HTTP Authorization Scheme:** `basic`

---

## resource-api-key

Authenticate with resource-specific API Keys using HTTP Basic Auth. Treat the resource-specific API Key ID as the username and resource-specific API Key Secret as the password.

**Security Scheme Type:** HTTP

**HTTP Authorization Scheme:** `basic`

---

## external-access-token

Authenticate with OAuth 2.0.

**Security Scheme Type:** OAuth2

**Flow type:** clientCredentials

**Token URL:** https://<oauth-identity-provider>/token

## oauth

Authenticate with OAuth 2.0. Currently this is only supported for partner APIs.

**Security Scheme Type:** OAuth2

**Flow type:** clientCredentials

**Token URL:** /oauth2/token

**Scopes:**

- partner:alter - enables partners to alter entitlements
  - partner:create - enables partners to create entitlements and sign on behalf of customers
- [See more](#)

## Errors

Note

This section describes the structure of error responses for many Confluent Cloud APIs, but not all. The Connect v1 API group has a different set of structures for error responses. Please review the example request and response bodies in the Connect v1 API documentation [below](#) to see its error behaviour.

Confluent uses conventional [HTTP status codes](#) to indicate the success or failure of an API request.

Failures follow a standard model to tell you about what went wrong. They may include one or more error objects with the following fields:

Field	Type	Description
id*	UUID	A unique identifier for this particular occurrence of the problem.
status	String	The HTTP status code applicable to this problem.
code	String	An application-specific error code.
title	String	A short, human-readable summary of the problem that should not change from occurrence to occurrence of the problem, except for purposes of localization.
detail*	String	A human-readable explanation specific to this occurrence of the problem. Like title, this field's value can be localized.
source	Object	An object that references the source of the error, and optionally includes any of the following members:
pointer	String	A <a href="#">JSON Pointer</a> to the associated entity in the request document (e.g. <code>"/spec/title"</code> for a specific attribute).
parameter	String	A string indicating which URI query parameter caused the error.
meta	Object	A meta object that contains non-standard meta-information about the error.

resolution	String	Instructions for the end-user for correcting the error.
------------	--------	---

\* indicates a required field

All errors include an `id` and some `detail` message. The `id` is a unique identifier — use it when you're working with Confluent support to debug a problem with a specific API call. The `detail` describes what went wrong.

Some errors that could be handled programmatically (e.g., a Kafka cluster config is invalid) may include an error `code` that briefly explains the error reported.

Validation issues and similar errors include a `source` which tells you exactly what in the request was responsible for the error.

For example, a failure may look like

```
{
  "errors": [
    {
      "status": "422",
      "code": "invalid_configuration",
      "id": "30ce6058-87da-11e4-b116-123b93f75cba",
      "title": "The Kafka cluster configuration is invalid",
      "detail": "The property '/cluster/storage_size' of type string did not match the expected type integer",
      "source": {
        "pointer": "/cluster/storage_size"
      }
    }
  ]
}
```

If a request fails validation, it will return an HTTP `422 Unprocessable Entity` with a list of fields that failed validation.

## Pagination

Note

This section describes the pagination behavior of “list” operations for many Confluent Cloud APIs, but not all. The Connect v1 API list operations do not support pagination.

All API resources have support for bulk reads via “list” API operations. For example, you can “list Kafka clusters”, “list api keys”, and “list environments”. These “list” operations require pagination; by requesting smaller subsets of data, API clients receive a response much faster than requesting the entire, potentially large, data set.

All “list” operations follow the same pattern with the following parameters:

- `page_size` – client-provided max number of items per page, only valid on the first request.
- `page_token` – server-generated token used for traversing through the result set.

A paginated response may include any of the following pagination links. API clients may follow the respective link to page forward or backward through the result set as desired.

Link Relation	Description
<code>next</code>	A link to the next page of results. A response that does not contain a next link does not have further data to fetch.
<code>prev</code>	A link to the previous page of results. A response that does not contain a prev link has no previous data. This link is optional for collections that cannot be traversed backward.
<code>first</code>	A link to the first page of results. This link is optional for collections that cannot be indexed directly to a given page.
<code>last</code>	A link to the last page of results. This link is optional for collections that cannot be indexed directly to a given page.

API clients must treat pagination links and the `page_token` parameter in particular as an opaque string.

An example paginated list response may look like

```
{
  "api_version": "v2",
  "kind": "KafkaClusterList",
  "metadata": {
    "next": "https://api.confluent.cloud/kafka-clusters?page_token=ABCDE"
  },
  "data": [
    {
      "metadata": {
        "id": "lkc-abc123",
        "self": "https://api.confluent.cloud/kafka-clusters/lkc-abc123",
        "resource_name": "crn://confluent.cloud/kafka=lkc-abc123",
      },
      "spec": {
        "display_name": "My Kafka Cluster",
        <snip>
      },
      "status": {
        "phase": "RUNNING",
        <snip>
      }
    },
    <snip>
  ]
}
```

## Rate Limiting

To protect the stability of the API and keep it available to all users, Confluent employs multiple safeguards. If you send too many requests in quick succession or perform too many concurrent operations, you may be throttled or have your request rejected with an error.

When a rate limit is breached, an HTTP `429 Too Many Requests` error is returned. The following headers are sent back to provide assistance in dealing with rate limits. Note that headers are not returned for a `429` error response with [Kafka REST API \(v3\)](#).

Header	Description
<code>X-RateLimit-Limit</code>	The maximum number of requests you're permitted to make per time period.
<code>X-RateLimit-Reset</code>	The relative time in seconds until the current rate limit window resets.
<code>Retry-After</code>	The number of seconds to wait until the rate limit window resets. Only sent when the rate limit is reached.
<code>X-RateLimit-Remaining</code>	The number of requests remaining in the current rate-limit window. Important: This differs from Github and Twitter's same-named header which uses UTC epoch seconds. We use relative time to avoid client/server time synchronization issues.

Confluent enforces multiple kinds of limits, including request-rate and concurrency limits, both per user and organization-wide. Unauthenticated requests are associated with the originating IP address, not the user making requests.

Integrations should gracefully handle these limits by watching for `429` error responses and building in a retry mechanism. This mechanism should follow a capped exponential backoff policy to prevent [retry amplification](#) ("retry storms") and also introduce some randomness ("jitter") to avoid the [thundering herd effect](#).

If you're running into this error and think you need a higher rate limit, contact Confluent at [support@confluent.io](mailto:support@confluent.io).

## Identifiers and URLs

Most resources have multiple identifiers:

- `id` is the "natural identifier" for an object. It is only unique within its parent resource. The `id` is unique across time: the ID will not be reclaimed and reused after an object is deleted.
- `resource_name` is a Uniform Resource Identifier (URI) that is globally unique across all resources. This encompasses all parent resource `kind`s and `id`s necessary to uniquely identify a particular instance of this object `kind`. Because it uses object `id`s, the CRN will not be reclaimed and reused after an object is deleted. It is represented as a Confluent Resource Name (see below).
- `self` is a Uniform Resource Locator (URL) at which an object can be addressed. This URL encodes the service location, API version, and other particulars necessary to locate the resource at a point in time.

To see how these relate to each other, consider `KafkaBroker` with `broker.id=2` in a `KafkaCluster` in Confluent Cloud identified as `lkc-xsi8201`. In such an example, the `KafkaBroker` has `id=2`, the `resource_name` is `crn://confluent.cloud/kafka=lkc-xsi8201/broker=2` and the `self` URL may be something like `https://pkc-8wlk2n.us-west-2.aws.confluent.cloud`. Note that different identifiers carry different information for different purposes, but the `resource_name` is the most complete and canonical identifier.

### Confluent Resource Names (CRNs)

*Confluent Resource Names* (CRNs) are used to uniquely identify all Confluent resources.

A CRN is a valid URI having an "authority" of `confluent.cloud` or a self-managed [metadata service URL](#), followed by the minimal hierarchical set of key-value pairs necessary to uniquely identify a resource.

Here are some examples for basic resources in Confluent Cloud:

Resource	Example CRN
Organization	<code>crn://confluent.cloud/organization=9bb441c4-edef-46ac-8a41-c49e44a3fd9a</code>
Environment	<code>crn://confluent.cloud/organization=9bb441c4-edef-46ac-8a41-c49e44a3fd9a/environment=env-456xy</code>
User	<code>crn://confluent.cloud/organization=9bb441c4-edef-46ac-8a41-c49e44a3fd9a/user=u-rst9876</code>
API Key	<code>crn://confluent.cloud/organization=9bb441c4-edef-46ac-8a41-c49e44a3fd9a/user=u-zyx98/api-key=ABCDEFGG9876543210</code>
Service Account	<code>crn://confluent.cloud/organization=9bb441c4-edef-46ac-8a41-c49e44a3fd9a/service-account=sa-abc1234</code>
Kafka Cluster	<code>crn://confluent.cloud/organization=9bb441c4-edef-46ac-8a41-c49e44a3fd9a/environment=env-456xy/cloud-cluster=lkc-123abc/kafka=lkc-123abc</code>
Kafka Topic	<code>crn://confluent.cloud/organization=9bb441c4-edef-46ac-8a41-c49e44a3fd9a/environment=env-456xy/cloud-cluster=lkc-123abc/kafka=lkc-123abc/topic=my_kafka_topic</code>
Consumer Group	<code>crn://confluent.cloud/organization=9bb441c4-edef-46ac-8a41-c49e44a3fd9a/environment=env-456xy/cloud-cluster=lkc-123abc/kafka=lkc-123abc/group=confluent_cli_consumer_123</code>
Network	<code>crn://confluent.cloud/organization=9bb441c4-edef-46ac-8a41-c49e44a3fd9a/environment=env-456xy/network=n-123abc</code>
Peering	<code>crn://confluent.cloud/organization=9bb441c4-edef-46ac-8a41-c49e44a3fd9a/environment=env-456xy/network=n-123abc/peering=p-123abc</code>
Private Link Access	<code>crn://confluent.cloud/organization=9bb441c4-edef-46ac-8a41-c49e44a3fd9a/environment=env-456xy/network=n-123abc/private-link-access=pla-123abc</code>
Transit Gateway	<code>crn://confluent.cloud/organization=9bb441c4-edef-46ac-8a41-c49e44a3fd9a/environment=env-456xy/network=n-</code>

Attachment	123abc/transit-gateway-attachment=tgwa-123abc
Schema Registry Cluster	crn://confluent.cloud/organization=9bb441c4-edef-46ac-8a41-c49e44a3fd9a/environment=env-456xy/schema-registry=lsrc-789qw
Schema Subject	crn://confluent.cloud/organization=9bb441c4-edef-46ac-8a41-c49e44a3fd9a/environment=env-456xy/schema-registry=lsrc-789qw/subject=test
Connector	crn://confluent.cloud/organization=9bb441c4-edef-46ac-8a41-c49e44a3fd9a/environment=env-456xy/cloud-cluster=lkc-123abc/connector=my_datagen_connector
Provider Integration	crn://confluent.cloud/organization=9bb441c4-edef-46ac-8a41-c49e44a3fd9a/environment=env-456xy/provider-integration=cspi-123j1

## Data Types

### Primitive Types

Data Type	Representation
Integers	Each API may specify the type as <code>int32</code> or <code>int64</code> . Note that many languages, including JavaScript, are limited to a max size of approx <code>2**53</code> and don't correctly handle large <code>int64</code> values with their default JSON parser.
Dates	<a href="#">RFC 3339</a> formatted string. UTC timezones are assumed, unless otherwise given.
Times	<a href="#">RFC 3339</a> formatted string. UTC timezones are assumed, unless otherwise given.
Durations	<a href="#">RFC 3339</a> formatted string.
Periods	<a href="#">RFC 3339</a> formatted string. UTC timezones are assumed, unless otherwise given.
Ranges	All ranges are represented using half-open intervals with naming conventions like <code>[start_XXX, end_XXX)</code> such as <code>[start_time, end_time)</code> .
Enums	Most APIs use <code>x-extensible-enum</code> as an open-ended list of values. This improves compatibility compared with a standard <code>enum</code> which by definition represents a closed set. All enums have a <code>0</code> -valued entry which either serves as the default for common cases, or represents <code>UNSPECIFIED</code> when no default exists and results in an error.

### Standard Properties

Confluent uses this set of standard properties to ensure common concepts use the same name and semantics across different APIs.

Name	Description
api_version	Many API objects have an <code>api_version</code> field indicating their API version. See the <a href="#">Object Model</a> .
kind	Many API objects have a <code>kind</code> field indicating the kind of object it is. See the <a href="#">Object Model</a> .
id	Many objects in the API will have an identifier, indicated via its <code>id</code> field, and should be treated as an opaque string unless otherwise specified. See the <a href="#">Object Model</a> .
name	Objects which support a client-provided unique identifier instead of a generated <code>id</code> will indicate this identifier via its <code>name</code> field.
display_name	The human-readable display name of an API object.
title	The official name of an API object, such as a company name. It should be treated as the formal version



	of <code>display_name</code> .
description	One or more paragraphs of text description of an entity.
created_at	The date and time the object was created, represented as a string in <a href="#">RFC 3339</a> format.
updated_at	The date and time the object was last modified, represented as a string in <a href="#">RFC 3339</a> format.
deleted_at	If present, the date and time after which the object was/will be deleted, represented as a string in <a href="#">RFC 3339</a> format.
page_token	The pagination token in the List request. See <a href="#">Pagination</a> .
page_size	The pagination size in the List request. See <a href="#">Pagination</a> .
total_size	The total count of items in the list irrespective of pagination. See <a href="#">Pagination</a> .
spec	The <i>desired state</i> specification of the resource, as observed by Confluent Cloud.
status	The <i>current state</i> of the resource, as observed by Confluent Cloud.

## Versioning

Confluent APIs ensure stability for your integrations by avoiding the introduction of breaking changes to customers unexpectedly. Confluent will make non-breaking API changes without advance notice. Thus, API clients must follow the [Compatibility Policy](#) below to ensure your integration remains stable. All APIs follow the API Lifecycle Policy described below, which describes the guarantees API clients can rely on.

Breaking changes will be [widely communicated](#) in advance in accordance with the Confluent [Deprecation Policy](#). Confluent will provide timelines and a migration path for all API changes, where available. Be sure to subscribe to one or more [communication channels](#) so you don't miss any updates!

One exception to these guidelines is for critical security issues. Confluent will take any necessary actions to mitigate any critical security issue as soon as possible, which may include disabling the vulnerable functionality until a proper solution is available.

Do not consume any Confluent API unless it is documented in the API Reference. All undocumented endpoints should be considered private, subject to change without notice, and not covered by any agreements.

Note: The version in the URL (e.g. "v1" or "v2") is not a "major version" in the [Semantic Versioning](#) sense. It is a "generational version" or "meta version", as seen in APIs like [Github API](#) or the [Stripe API](#).

## API Groups

Confluent APIs are divided into API Groups, such as the Cluster Management for Apache Kafka (CMK) API group, the Connect API group, and the Data Catalog API group. Each group has its own set of endpoints and resources, as well as its own API group version.

Because different API groups have different versions, there is no single version for the "Confluent Cloud API". The latest version of the Connect API group may be `connect/v1`, while the latest version of the CMK API group may be `cmk/v2`.

When a breaking change is introduced into one API group, Confluent will increase the API version for that API group only, leaving the other API groups' versions unchanged. This makes it easier for you to understand whether a given breaking change impacts your usage of the APIs.

## Known Issues

During the Early Access and Preview periods, we have a few known issues.

Issue	Description	Proposed Resolution
-------	-------------	---------------------

Quota Exceeded	Some "Quota Exceeded" errors will be returned as HTTP 400 instead of HTTP 402	Return 402 consistently for "Quota Exceeded" errors
----------------	---	---

## API Lifecycle Policy

The following status labels are applicable to APIs, features, and SDK versions, based on the current support status of each:

- Early Access – May change at any time. Not recommended for production usage. Not officially supported by Confluent. Intended for user feedback only. Users must be granted explicit access to the API by Confluent.
- Preview – Unlikely to change between Preview and General Availability. Not recommended for production usage. Officially supported by Confluent for non-production usage. Accessible to all users.
- Limited Availability (LA) - Available to key select customers in a subset of regions/providers/networks and recommended for production usage.
- Generally Available (GA) – Will not change at short notice. Recommended for production usage. Officially supported by Confluent for non-production and production usage.
- Deprecated – Still supported, but no longer under active development. Existing usage will continue to function but migration following the upgrade guide is strongly recommended. New use cases should be built against the new version. Deprecated feature or version will be removed in the future at the announced date.
- Sunset – Removed, and no longer supported or available.

An API is "Generally Available" unless explicitly marked otherwise.

## Compatibility Policy

Confluent Cloud APIs are governed by [Confluent Cloud Upgrade Policy](#), which means that backward incompatible changes and deprecations will be made approximately once per year, and 180 days notice will be provided via email to all registered Confluent Cloud users.

### Backward Compatibility

*An API version is backward compatible if a program written against the previous version of the API will continue to work the same way, without modification, against this version of the API.*

Confluent considers the following changes to be backward compatible:

- Adding new API resources.
- Adding new optional parameters to existing API requests (e.g., query string).
- Adding new properties to existing API resources (e.g., request body).
- Changing the order of properties in existing API responses.
- Changing the length or format of object IDs or other opaque strings.
  - Unless otherwise documented, you can safely assume object IDs generated by Confluent will never exceed 255 characters, but you should be able to handle IDs of up to that length. If you're using MySQL, for example, you should store IDs in a `VARCHAR(255) COLLATE utf8_bin` column.
  - This includes adding or removing fixed prefixes (such as `lkc-` on Kafka cluster IDs).
  - This includes API keys, API tokens, and similar authentication mechanisms.
  - This includes all strings described as "opaque" in the docs, such as pagination cursors.
- Adding new API event types.
- Adding new properties to existing API event types.
- Omitting properties with null values from existing API responses.

### Forward Compatibility

*An API version is forward compatible if a program written against the next version of the API will continue to work the same way, without modification, against this version of the API.*

In other words, a forward compatible API will accept input intended for a later version of itself.

Confluent does not guarantee the forward compatibility of the APIs, but Confluent does generally follow the guidelines given by the [Robustness principle](#). This means that the API determines what to do with a request based only on the parts that it recognizes.

This is often referred to as the MUST IGNORE rule.

- Request parameters that are not recognized will be ignored (e.g., query string).
- Request properties that are not recognized will be ignored (e.g., request body).
- Request metadata that are not recognized will be ignored (e.g., request headers).

API clients must also follow the MUST IGNORE rule.

- Response properties that are not recognized must be ignored (e.g., response body).
- Response metadata that are not recognized must be ignored (e.g., response headers).

Additionally, there is a more subtle related rule called the MUST FORWARD rule. Any parts of a request that an API doesn't recognize must be forwarded unchanged.

- Response properties that are not recognized must be included in any input subsequent updates (e.g., request body)
  - This includes future `PUT` requests in a read/modify/write operation. (This isn't required for `PATCH` partial updates, which is why Confluent APIs use `PATCH`.)
- Event processors must not strip unknown properties before forwarding messages.

## Client Responsibilities

- Resource and rate limits, and the default and maximum sizes of paginated data are not considered part of the API contract and may change (possibly dynamically). It is the client's responsibility to read the road signs and obey the speed limit.
- If a property has a primitive type and the API documentation does not explicitly limit its possible values, clients must not assume the values are constrained to a particular set of possible responses.
- If a property of an object is not explicitly declared as mandatory in the API, clients must not assume it will be present.
- A resource may be modified to return a "redirection" response (e.g. `301`, `307`) instead of directly returning the resource. Clients must handle HTTP-level redirects, and respect HTTP headers (e.g. `Location`).

## Deprecation Policy

Confluent will announce deprecations at least 180 days in advance of a breaking change and will continue to maintain the deprecated APIs in their original form during this time.

Exceptions to this policy apply in case of critical security vulnerabilities or functional defects.

## Communication

When a deprecation is announced, the details and any relevant migration information will be available on one or more of the following channels:

- Announcements on the [Developer Blog](#), [Community Slack \(join!\)](#), [Google Group](#), the [@ConfluentInc](#) twitter account, and similar channels
- Enterprise customers may receive information by email to their specified Confluent contact, if applicable.

## HTTP Guidelines

## Status Codes

Confluent respects the meanings and behavior of HTTP status codes as defined in [RFC2616](#) and elsewhere.

- Codes in the `2xx` range indicate success
- Codes in the `3xx` range indicate redirection
- Codes in the `4xx` range indicate an error caused by the client request (e.g., a required parameter was omitted, an invalid cluster configuration was provided, etc.)
- Codes in the `5xx` range indicate an error with Confluent's servers (these are rare)

The various HTTP status codes that might be returned are listed below.

Code	Title	Description
200	OK	Everything worked as expected.
201	Created	The resource was created. Follow the <code>Location</code> header.
204	No Content	Everything worked and there is no content to return.
400	Bad Request	The request was unacceptable, often due to malformed syntax, or a missing or malformed parameter.
401	Unauthorized	No valid credentials provided. or the credentials are unsuitable, invalid, or unauthorized.
402	Over Quota	The request was valid, but you've exceeded your plan quota or limits.
404	Not Found	The requested resource doesn't exist or you're unauthorized to know it exists.
409	Conflict	The request conflicts with another request (perhaps it already exists or was based on a stale version of data).
422	Validation Failed	The request was parsed correctly but failed some sort of validation.
429	Too Many Requests	Too many requests hit the API too quickly. Confluent recommends an exponential backoff of your requests.
500, 502, 503, 504	Server Errors	Something went wrong on Confluent's end. (These are rare.)

This list is not exhaustive; other standard HTTP error codes may be used, including `304`, `307`, `308`, `405`, `406`, `408`, `410`, and `415`.

For more details, see <https://httpstatuses.com>.

## Metrics APIs

For Metrics APIs, see [Confluent Cloud Metrics API](#).

## API Keys (iam/v2)

Lifecycle Stage **General Availability**

`ApiKey` objects represent access to different parts of Confluent Cloud. Some types of API keys represent access to a single cluster/resource such as a Kafka cluster, Schema Registry cluster or a ksqldb cluster. Cloud API Keys represent access to resources within an organization that are not tied to a specific cluster, such as the Org API, IAM API, Metrics API or Connect API.

The API allows you to list, create, update and delete your API Keys.

Related guide: [API Keys in Confluent Cloud](#).

## The API Keys Model

api_version	string Value: <code>"iam/v2"</code> APIVersion defines the schema version of this representation of a resource.
kind	string Value: <code>"ApiKey"</code> Kind defines the object this REST resource represents.
id	string <code>&lt;= 255 characters</code> ID is the "natural identifier" for an object within its scope/namespace; it is normally unique across time but not space. That is, you can assume that the ID will not be reclaimed and reused after an object is deleted ("time"); however, it may collide with IDs for other object <code>kinds</code> or objects of the same <code>kind</code> within a different scope/namespace ("space").
metadata	object ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object (iam.v2.ApiKeySpec) The desired state of the Api Key

## Quotas and Limits

This resource is subject to the [following quotas](#):

Quota	Description
<code>apikeykeys_per_org</code>	API Keys in one Confluent Cloud organization

## List of API Keys

Lifecycle Stage **General Availability**

Retrieve a sorted, filtered, paginated list of all API keys.

This can show all keys for a single owner (across resources - Kafka clusters), or all keys for a single resource (across owners). If no `owner` or `resource` filters are specified, returns all API Keys in the organization. You will only see the keys that are accessible to the account making the API request.

AUTHORIZATIONS: `cloud-api-key`  
>

### QUERY PARAMETERS

spec.owner	string (SearchFilter) Filter the results by exact match for spec.owner.
spec.resource	string (SearchFilter) Filter the results by exact match for spec.resource.
page_size	integer <code>&lt;= 100</code> Default: <code>10</code> A pagination size for collection requests.
page_token	string <code>&lt;= 255 characters</code> An opaque pagination token for collection requests.

## Responses

> 200
API Key.

Copy	Expand all	Collapse all
<pre>{   "api_version": "iam/v2",   "kind": "ApiKey",   "id": "dlz-f3a90de",   "metadata": {     "self": "https://api.confluent.cloud",     "resource_name": "crn://confluent.cl",     "created_at": "2006-01-02T15:04:05-0",     "updated_at": "2006-01-02T15:04:05-0",     "deleted_at": "2006-01-02T15:04:05-0"   },   "spec": {     "secret": "Rl5hoiDIq8Nku/1Y4mP03DwAV",     "display_name": "CI kafka access key",     "description": "This API key provide",     "owner": { ... },     "resource": { ... }   } }</pre>		

### Request samples

Shell	Java	Go	Python	Node
C	C#			

Copy
<pre>curl --request GET \   --url 'https://api.confluent.cloud/iam/v2,   --header 'Authorization: Basic REPLACE_BA:</pre>

### Response samples

200	400	401	403	500
Content type application/json				
Copy	Expand all	Collapse all		
<pre>{   "api_version": "iam/v2",   "kind": "ApiKeyList",   "metadata": {     "first": "https://api.confluent.cloud",     "last": "https://api.confluent.cloud",     "prev": "https://api.confluent.cloud",     "next": "https://api.confluent.cloud",     "total_size": 123   },   "data": [     + { ... }   ] }</pre>				

> 400
Bad Request
> 401
Unauthorized
> 403
Forbidden
> 429
Rate Limit Exceeded
> 500
Oops, something went wrong!

## Create an API Key

Lifecycle Stage General Availability

Make a request to create an API key.

AUTHORIZATIONS: *cloud-api-key*

>

REQUEST BODY SCHEMA: *application/json*

spec > required  
object  
The desired state of the Api Key

## Responses

> 202

An API Key is being created.

> 400

Bad Request

> 401

Unauthorized

> 402

Over Quota

> 403

Forbidden

> 409

Conflict

> 422

Validation Failed

> 429

Rate Limit Exceeded

> 500

Oops, something went wrong!

## Read an API Key

}

POST /iam/v2/api-keys

### Request samples

Payload

Shell

Java

Go

Python

Node

C

C#

Content type  
application/json

Copy Expand all Collapse all

```
{
  - "spec": {
    "display_name": "CI kafka access key"
    "description": "This API key provide
  + "owner": { ... },
  + "resource": { ... }
  }
}
```

### Response samples

202

400

401

402

403

409

422

500

Content type  
application/json

Copy Expand all Collapse all

```
{
  "api_version": "iam/v2",
  "kind": "ApiKey",
  "id": "dlz-f3a90de",
  - "metadata": {
    "self": "https://api.confluent.cloud
    "resource_name": "crn://confluent.cl
    "created_at": "2006-01-02T15:04:05-0
    "updated_at": "2006-01-02T15:04:05-0
    "deleted_at": "2006-01-02T15:04:05-0
  },
  - "spec": {
    "secret": "R15hoiDIq8Nku/1Y4mP03DwAV
    "display_name": "CI kafka access key
    "description": "This API key provide
  + "owner": { ... },
  + "resource": { ... }
  }
}
```

GET /iam/v2/api-keys/{id}

Lifecycle Stage General Availability

Make a request to read an API key.

AUTHORIZATIONS: cloud-api-key  
>

PATH PARAMETERS

id string  
required The unique identifier for the API key.

## Responses

> 200
API Key.
> 400
Bad Request
> 401
Unauthorized
> 403
Forbidden
> 404
Not Found
> 429
Rate Limit Exceeded
> 500
Oops, something went wrong!

## Update an API Key

Lifecycle Stage General Availability

Make a request to update an API key.

AUTHORIZATIONS: cloud-api-key  
>

PATH PARAMETERS

id string  
required The unique identifier for the API key.

REQUEST BODY SCHEMA: application/json

spec > object (iam.v2.ApiKeySpecUpdate)  
The desired state of the Api Key

## Responses

> 200
API Key.
> 400
Bad Request

## Request samples

Shell Java Go Python Node  
C C#

Copy  
curl --request GET \\\n--url 'https://api.confluent.cloud/iam/v2,\n--header 'Authorization: Basic REPLACE\_BA!

## Response samples

200 400 401 403 404  
500

Content type  
application/json  
Copy Expand all Collapse all  
{\n "api\_version": "iam/v2",\n "kind": "ApiKey",\n "id": "dlz-f3a90de",\n - "metadata": {\n "self": "https://api.confluent.cloud\n "resource\_name": "crn://confluent.cl\n "created\_at": "2006-01-02T15:04:05-0\n "updated\_at": "2006-01-02T15:04:05-0\n "deleted\_at": "2006-01-02T15:04:05-0\n },\n - "spec": {\n "secret": "Rl5hoiDIq8Nku/1Y4mP03DwAV\n "display\_name": "CI kafka access key\n "description": "This API key provide\n + "owner": { ... },\n + "resource": { ... }\n }\n}

PATCH /iam/v2/api-keys/{id}

## Request samples

Payload Shell Java Go  
Python Node C C#

Content type  
application/json  
Copy Expand all Collapse all  
{\n - "spec": {\n "display\_name": "CI kafka access key\n "description": "This API key provide\n }\n}

## Response samples

200 400 401 402 403  
404 409 422 500

Content type  
application/json

> 401
Unauthorized
> 402
Over Quota
> 403
Forbidden
> 404
Not Found
> 409
Conflict
> 422
Validation Failed
> 429
Rate Limit Exceeded
> 500
Oops, something went wrong!

Copy Expand all Collapse all

```
{
  "api_version": "iam/v2",
  "kind": "ApiKey",
  "id": "dlz-f3a90de",
  - "metadata": {
    "self": "https://api.confluent.cloud",
    "resource_name": "crn://confluent.cl",
    "created_at": "2006-01-02T15:04:05-0",
    "updated_at": "2006-01-02T15:04:05-0",
    "deleted_at": "2006-01-02T15:04:05-0"
  },
  - "spec": {
    "secret": "Rl5hoiDIq8Nxu/1Y4mPO3DwAV",
    "display_name": "CI kafka access key",
    "description": "This API key provide",
    + "owner": { ... },
    + "resource": { ... }
  }
}
```

## Delete an API Key

Lifecycle Stage General Availability

Make a request to delete an API key.

AUTHORIZATIONS: *cloud-api-key*

>

PATH PARAMETERS

id string required  
The unique identifier for the API key.

### Responses

> 204
An API Key is being deleted.
> 400
Bad Request
> 401
Unauthorized
> 403
Forbidden
> 404
Not Found
> 429
Rate Limit Exceeded
> 500
Oops, something went wrong!

DELETE /iam/v2/api-keys/{id}

### Request samples

Shell Java Go Python Node  
C C#

Copy

```
curl --request DELETE \
  --url 'https://api.confluent.cloud/iam/v2,
  --header 'Authorization: Basic REPLACE_BA'
```

### Response samples

400 401 403 404 500

Content type  
application/json

Copy Expand all Collapse all

```
{
  - "errors": [
    + { ... }
  ]
}
```

## Users (iam/v2)

Lifecycle Stage General Availability

User objects represent individuals who may access your Confluent resources.



The API allows you to retrieve, update, and delete individual users, as well as list of all your users. This API cannot be used to create new user accounts.

Related guide: [Users in Confluent Cloud](#).

The Users Model

api_version	string Value: <code>"iam/v2"</code> APIVersion defines the schema version of this representation of a resource.
kind	string Value: <code>"User"</code> Kind defines the object this REST resource represents.
id	string ≤ 255 characters ID is the "natural identifier" for an object within its scope/namespace; it is normally unique across time but not space. That is, you can assume that the ID will not be reclaimed and reused after an object is deleted ("time"); however, it may collide with IDs for other object <code>kinds</code> or objects of the same <code>kind</code> within a different scope/namespace ("space").
metadata	object ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
email	string <email> The user's email address
full_name	string The user's full name
auth_type	string extensible-enum: <code>["AUTH_TYPE_LOCAL", "AUTH_TYPE_SSO"]</code> The user's authentication method

CopyExpand allCollapse all

```
{
  "api_version": "iam/v2",
  "kind": "User",
  "id": "dlz-f3a90de",
  "metadata": {
    "self": "https://api.confluent.cloud",
    "resource_name": "crn://confluent.cl",
    "created_at": "2006-01-02T15:04:05-0",
    "updated_at": "2006-01-02T15:04:05-0",
    "deleted_at": "2006-01-02T15:04:05-0",
  },
  "email": "marty.mcfly@example.com",
  "full_name": "Marty McFly",
  "auth_type": "AUTH_TYPE_SSO"
}
```

Quotas and Limits

This resource is subject to the [following quotas](#):

Quota	Description
<code>users_per_org</code>	Users in one Confluent Cloud organization

List of Users

Lifecycle StageGeneral Availability

Retrieve a sorted, filtered, paginated list of all users.

AUTHORIZATIONS: `cloud-api-key` or `confluent-sts-access-token`

QUERY PARAMETERS

page_size	integer ≤ 100 Default: <code>10</code> A pagination size for collection requests.
page_token	string ≤ 255 characters An opaque pagination token for collection requests.

Responses

GET /iam/v2/users

Request samples

ShellJavaGoPythonNodeC C#

Copy

```
curl --request GET \
  --url https://api.confluent.cloud/iam/v2/
  --header 'Authorization: Basic REPLACE_BA'
```

Response samples

200400401403500

Content type

application/json

> 200
User.
> 400
Bad Request
> 401
Unauthorized
> 403
Forbidden
> 429
Rate Limit Exceeded
> 500
Oops, something went wrong!

CopyExpand allCollapse all

```
{
  "api_version": "iam/v2",
  "kind": "UserList",
  - "metadata": {
    "first": "https://api.confluent.cloud"
    "last": "https://api.confluent.cloud"
    "prev": "https://api.confluent.cloud"
    "next": "https://api.confluent.cloud"
    "total_size": 123
  },
  - "data": [
    + { ... }
  ]
}
```

## Read a User

Lifecycle StageGeneral Availability

Make a request to read a user.

AUTHORIZATIONS: *cloud-api-key* or *confluent-sts-access-token*

### PATH PARAMETERS

id	string
required	The unique identifier for the user.

## Responses

> 200
User.
> 400
Bad Request
> 401
Unauthorized
> 403
Forbidden
> 404
Not Found
> 429
Rate Limit Exceeded
> 500
Oops, something went wrong!

GET /iam/v2/users/{id}

### Request samples

ShellJavaGoPythonNodeC C#

Copy

```
curl --request GET \
  --url 'https://api.confluent.cloud/iam/v2/'
  --header 'Authorization: Basic REPLACE_BA'
```

### Response samples

200400401403404500

Content typeapplication/json

CopyExpand allCollapse all

```
{
  "api_version": "iam/v2",
  "kind": "User",
  "id": "dlz-f3a90de",
  - "metadata": {
    "self": "https://api.confluent.cloud"
    "resource_name": "crn://confluent.cl"
    "created_at": "2006-01-02T15:04:05-0"
    "updated_at": "2006-01-02T15:04:05-0"
    "deleted_at": "2006-01-02T15:04:05-0"
  },
  "email": "marty.mcfly@example.com",
  "full_name": "Marty McFly",
  "auth_type": "AUTH_TYPE_SSO"
}
```

## Update a User

Lifecycle StageGeneral Availability

Make a request to update a user.

AUTHORIZATIONS: *cloud-api-key* or *confluent-sts-access-token*

### PATH PARAMETERS

id	string
----	--------

PATCH /iam/v2/users/{id}

### Request samples

PayloadShellJavaGoPythonNodeC C#

Content typeapplication/json

required

The unique identifier for the user.

REQUEST BODY SCHEMA: application/json

full\_name

string

The user's full name

Responses

> 200
User.
> 400
Bad Request
> 401
Unauthorized
> 402
Over Quota
> 403
Forbidden
> 404
Not Found
> 409
Conflict
> 422
Validation Failed
> 429
Rate Limit Exceeded
> 500
Oops, something went wrong!

Delete a User

Lifecycle Stage General Availability

Make a request to delete a user.

If successful, this request will also recursively delete all of the user's associated resources, including its cloud and cluster API keys.

AUTHORIZATIONS: cloud-api-key or confluent-sts-access-token  
>

PATH PARAMETERS

id

required

string

The unique identifier for the user.

Responses

> 204
A User is being deleted.
> 400
Bad Request
> 401
Unauthorized

Copy

{  
  
 "full\_name": "Marty McFly"  
}

Response samples

200

400

401

402

403

404

409

422

500

Content type

application/json

Copy

Expand all

Collapse all

{  
  
 "api\_version": "iam/v2",  
 "kind": "User",  
 "id": "dlz-f3a90de",  
 - "metadata": (  
 "self": "https://api.confluent.cloud  
 "resource\_name": "crn://confluent.cl  
 "created\_at": "2006-01-02T15:04:05-0  
 "updated\_at": "2006-01-02T15:04:05-0  
 "deleted\_at": "2006-01-02T15:04:05-0  
 ),  
 "email": "marty.mcfly@example.com",  
 "full\_name": "Marty McFly",  
 "auth\_type": "AUTH\_TYPE\_SSO"  
}

DELETE

/iam/v2/users/{id}

Request samples

Shell

Java

Go

Python

Node

C

C#

Copy

curl --request DELETE \  
  
 --url 'https://api.confluent.cloud/iam/v2,  
 --header 'Authorization: Basic REPLACE\_BA!

Response samples

400

401

403

404

500

Content type

application/json

Copy

Expand all

Collapse all

{  
  
 - "errors": (  
 + { \_ }  
 )  
}

> 403
Forbidden
> 404
Not Found
> 429
Rate Limit Exceeded
> 500
Oops, something went wrong!

## Service Accounts (iam/v2)

Lifecycle Stage **General Availability**

`ServiceAccount` objects are typically used to represent applications and other non-human principals that may access your Confluent resources.

The API allows you to create, retrieve, update, and delete individual service accounts, as well as list all your service accounts.

Related guide: [Service Accounts in Confluent Cloud](#).

### The Service Accounts Model

api_version	string Value: <code>"iam/v2"</code> APIVersion defines the schema version of this representation of a resource.
kind	string Value: <code>"ServiceAccount"</code> Kind defines the object this REST resource represents.
id	string <code>&lt;= 255 characters</code> ID is the "natural identifier" for an object within its scope/namespace; it is normally unique across time but not space. That is, you can assume that the ID will not be reclaimed and reused after an object is deleted ("time"); however, it may collide with IDs for other object <code>kinds</code> or objects of the same <code>kind</code> within a different scope/namespace ("space").
metadata >	object ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
display_name	string A human-readable name for the Service Account
description	string A free-form description of the Service Account

### Quotas and Limits

This resource is subject to the [following quotas](#):

Quota	Description
<code>service_accounts_per_organization</code>	Service Accounts in one Confluent Cloud organization

## List of Service Accounts

Lifecycle Stage **General Availability**

Copy Expand all Collapse all

```
{
  "api_version": "iam/v2",
  "kind": "ServiceAccount",
  "id": "dlz-f3a90de",
  "metadata": {
    "self": "https://api.confluent.cloud",
    "resource_name": "crn://confluent.cl",
    "created_at": "2006-01-02T15:04:05-0",
    "updated_at": "2006-01-02T15:04:05-0",
    "deleted_at": "2006-01-02T15:04:05-0"
  },
  "display_name": "DeLorean_auto_repair",
  "description": "Doc's repair bot for the"
}
```

GET /iam/v2/service-accounts