DOCUMENTATION

Duo Admin API

Last Updated: October 31st, 2024

Contents

Related

The Duo Admin API provides programmatic access to the administrative functionality of Duo Security's two-factor authentication platform.

Overview

Duo Admin API is automatically available to paying Duo Premier, Duo Advantage, and Duo Essentials plan customers and new customers with an Advantage or Premier trial. Learn how to sign up for a Duo account and receive a free 30-day Duo Advantage trial.

The Admin API lets developers integrate with Duo Security's platform at a low level. The API has methods for creating, retrieving, updating, and deleting the core objects in Duo's system: users, phones, hardware tokens, admins, and integrations.

Developers can write applications that programmatically read their Duo account's authentication logs, administrator logs, and telephony logs; read or update account settings; and retrieve reports and other information.

Review the API Details to see how to construct your first API request.

> Are you a software vendor looking to integrate Duo into your application? Join our free Duo Technology Partner Program for developer accounts, joint marketing support, and more!

About the Admin API

Documented properties will not be removed within a stable version of the API. Once a given API endpoint is documented to return a given property, a property with that name will always appear (although certain properties may only appear under certain conditions, like if the customer is using a specific edition). When Duo deprecates a property, the API continues to accept that property in requests, although it no longer has any effect.

Properties that enumerate choices may gain new values at any time, e.g. the device `platform` value could return new device platforms that did not previously exist. Duo may cease to return legacy values for properties as well. Duo will update our API documentation with changes to property values in a timely fashion, adding new property values or indicating changes to existing property values.

New, undocumented properties may also appear at any time. For instance, Duo may make available a beta feature involving extra information returned by an API endpoint. Until the property is documented here its format may change or it may even be entirely removed from our API.

We have started implementing v2 handlers for endpoints. In these cases, the API v1 handler remains supported, but will be limited or deprecated in the future. We encourage use of the v2 endpoints where available and recommend migrating existing API implementations to the v2 handlers.

Please note that all Unix timestamps are in seconds, except where noted.

First Steps

Role required: Owner

> Note that only administrators with the **Owner** role can create or modify an Admin API application in the Duo Admin Panel.

1. Sign up for a Duo account if you aren't already a customer. Your free 30-day Duo Advantage trial includes Admin API access.
2. Log in to the Duo Admin Panel and navigate to **Applications** → **Protect an Application**.
3. Locate the entry for **Admin API** in the applications list. Click **Protect** to the far-right to configure the application and get your **integration key**, **secret key**, and **API hostname**. You'll need this information to complete your setup. See Protecting Applications for more information about protecting applications in Duo and additional application options.

> **Treat your secret key like a password**
>
> The security of your Duo application is tied to the security of your secret key (skey). Secure it as you would any sensitive credential. Don't share it with unauthorized individuals or email it to anyone under any circumstances!

Determine the permissions you want to grant to this Admin API application. Refer to the API endpoint descriptions throughout this document for information about required permissions for operations.

Permission	Details
Grant administrators - Read	The Admin API application can read information about Duo administrators and administrative units.
Grant administrators - Write	The Admin API application can read, add, modify, and delete information about Duo administrators and administrative units.
Grant read information	The Admin API application can read information about the Duo customer account's utilization.
Grant applications	The Admin API application can add, modify, and delete applications (referred to as "integrations" in the API), including permissions on itself or other Admin API applications.
Grant settings	The Admin API application can read and change global Duo account settings.
Grant read log	The Admin API application can read authentication, offline access, telephony, and administrator action log information.
Grant resource - Read	The Admin API application can read information about resource objects such as end users, policies, and devices.
Grant resource - Write	The Admin API application can create, update, and delete resource objects such as end users, policies, and devices.
Grant set Admin API permissions	The Admin API application can add or remove the permissions listed above via API for other Admin API applications. When this permission is not granted, permissions for an Admin API application must be set from the Duo Admin Panel.

Optionally specify which IP addresses or ranges are allowed to use this Admin API application in **Networks for API Access**. If you do not specify any IP addresses or ranges, this Admin API application may be accessed from any network.

The Admin API performs the IP check after verifying the authentication signature in a request. If you restrict the allowed networks for API access and see logged events for blocked Admin API requests from unrecognized IP addresses, this may indicate compromise of your Admin API application's secret key.

> **Connectivity Requirements**
> This application communicates with Duo's service on SSL TCP port 443.
>
> Firewall configurations that restrict outbound access to Duo's service with rules using destination IP addresses or IP address ranges aren't recommended, since these may change over time to maintain our service's high availability. If your organization requires IP-based rules, please review Duo Knowledge Base article 1337.
>
> Effective June 30, 2023, Duo no longer supports TLS 1.0 or 1.1 connections or insecure TLS/SSL cipher suites. See Duo Knowledge Base article 7546 for additional guidance.

API Clients

Duo Security has demonstration clients available on GitHub to call the Duo API methods.

- Python (duo_client_python)
- Java (duo_client_java)
- C# (duo_api_csharp)
- Go (duo_api_golang)
- Node (duo_api_nodejs)
- Ruby (duo_api_ruby)
- Perl (duo_api_perl)
- PHP (duo_api_php)

API Details

Base URL

All API methods use your **API hostname**, `https://api-XXXXXXXX.duosecurity.com`. Obtain this value from the Duo Admin Panel and use it exactly as shown there.

Methods always use HTTPS. Unsecured HTTP is not supported.

Request Format

All requests must have "Authorization" and "Date" headers.

If the request method is `GET` or `DELETE`, URL-encode parameters and send them in the URL query string like this: `/admin/v1/users?realname=First%20Last&username=root`. They still go on a separate line when creating the string to sign for an Authorization header.

Send parameters for `POST` requests in the body as URL-encoded key-value pairs (the same request format used by browsers to submit form data). The header "`Content-Type: application/x-www-form-urlencoded`" must also be present.

When URL-encoding, all bytes except ASCII letters, digits, underscore ("_"), period ("."), tilde ("~"), and hyphen ("-") are replaced by a percent sign ("%") followed by two hexadecimal digits containing the value of the byte. For example, a space is replaced with "%20" and an at-sign ("@") becomes "%40". Use only upper-case A through F for hexadecimal digits.

A request with parameters, as a complete URL, would look like this: `https://api-XXXXXXXX.duosecurity.com/admin/v1/users?realname=First%20Last&username=root`.

Response Format

Responses are formatted as a JSON object with a top-level `stat` key.

Successful responses will have a `stat` value of "OK" and a `response` key. The `response` will either be a single object or a sequence of other JSON types, depending on which endpoint is called.

```
{
  "stat": "OK",
  "response": {
    "key": "value"
  }
}
```

Values are returned as strings unless otherwise documented.

Unsuccessful responses will have a `stat` value of "FAIL", an integer `code`, and a `message` key that further describes the failure. A `message_detail` key may be present if additional information is available (like the specific parameter that caused the error).

```
{
  "stat": "FAIL",
  "code": 40002,
  "message": "Invalid request parameters",
  "message_detail": "username"
}
```

The HTTP response code will be the first three digits of the more specific `code` found inside the JSON object. Each endpoint's documentation lists HTTP response codes it can return. Additionally, all API endpoints that require a signed request can return the following HTTP response codes:

Response	Meaning
200	The request completed successfully.
401	The "Authorization", "Date", and/or "Content-Type" headers were missing or invalid.
403	This integration is not authorized for this endpoint or the ikey was created for a different integration type (for example, using an Auth API ikey with Admin API endpoints).
405	The request's HTTP verb is not valid for this endpoint (for example, POST when only GET is supported).
429	The account has made too many requests of this type recently. Try again later.

Response Paging

Some API endpoints return a paged list of results on `GET`, up to the API endpoint's `limit`, or maximum results per page.

A successful response when the total results exceed the endpoint's default page size will include a metadata section with information about the total number of objects found and the results returned in the paged response. If the request returns no paging metadata, then either the endpoint does not support paged results or the total results do not exceed one page.

Specifying incorrect paging parameters results in a `400` invalid parameters response.

Metadata Information	Description
`total_objects`	An integer indicating the total number of objects retrieved by the API request across all pages of results.
`next_offset`	An integer indicating The offset from `0` at which to start the next paged set of results. If not present in the metadata response, then there are no more pages of results left.
`prev_offset`	An integer indicating the offset from `0` at which the previous paged set of results started. If you did not specify `next_offset` in the request, this defaults to `0` (the beginning of the results).

Use the metadata information returned to change the paging parameters for your request.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned in a paged set of results. Each endpoint that supports paged results has its own `limit` settings, specified like "Default: `100`; Max: `300`". If a request specifies a value greater than the endpoint's maximum `limit`, max value is used.
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

To retrieve the full set of results for a request with paged results, repeat the call, specifying the `offset` parameter value, until there are no more results (indicated by the absence of `next_offset`).

Paging Metadata Examples:

The metadata response will look like these examples except where noted for an individual API endpoint.

This metadata information indicates that there are 951 total objects returned by that endpoint, and no `offset` or `limit` was specified so the response set defaulted to the first 100 objects:

```
{
    "metadata": {
        "next_offset": 100,
        "prev_offset": 0,
        "total_objects": 951
    }
```

This metadata information indicates that the request specified `offset=500 limit=200`, so the response set was objects 500-699:

```
{
    "metadata": {
        "next_offset": 700,
        "prev_offset": 300,
        "total_objects": 11318
    }
```

This metadata information indicates that there are 2342 total objects, and the request specified `offset=2300` and used that endpoint's default `limit` of 100, so the response set was the end of the list (objects 2300-2342):

```
{
    "metadata": {
        "prev_offset": 2200,
        "total_objects": 2342
    }
```

Authentication

The API uses HTTP Basic Authentication to authenticate requests. Use your Duo application's **integration key** as the HTTP Username.

Generate the HTTP Password as an HMAC signature of the request. This will be different for each request and must be re-generated each time.

To construct the signature, first build an ASCII string from your request, using the following components:

Component	Description	Example
`date`	The current time, formatted as RFC 2822. This must be the same string as the "Date" header.	`Tue, 21 Aug 2012 17:29:18 -0000`
`method`	The HTTP method (uppercase)	`POST`
`host`	Your **API hostname** (lowercase)	`api-xxxxxxxx.duosecurity.com`
`path`	The specific API method's path	`/admin/v1/users`
`params`	The URL-encoded list of `key=value` pairs, lexicographically sorted by key. These come from the request parameters (the URL query string for GET and DELETE requests or the request body for POST requests). If the request does not have any parameters one must still include a blank line in the string that is signed. Do not encode unreserved characters. Use upper-case hexadecimal digits A through F in escape sequences.	An example `params` list: `realname=First%20Last&username=root`

Then concatenate these components with (line feed) newlines. For example:

```
Tue, 21 Aug 2012 17:29:18 -0000
POST
api-xxxxxxxx.duosecurity.com
/admin/v1/users
realname=First%20Last&username=root
```

GET requests also use this five-line format:

```
Tue, 21 Aug 2012 17:29:18 -0000
GET
api-xxxxxxxx.duosecurity.com
/admin/v1/users
username=root
```

Lastly, compute the HMAC-SHA1 of this canonical representation, using your Duo Admin API application's **secret key** as the HMAC key. Send this signature as hexadecimal ASCII (i.e. not raw binary data). Use HTTP Basic Authentication for the request, using your **integration key** as the username and the HMAC-SHA1 signature as the password. Signature validation is case-insensitive, so the signature may be upper or lowercase.

For example, here are the headers for the above POST request to `api-XXXXXXXX.duosecurity.com/admin/v1/users`, using `DIWJ8X6AEYOR5OMC6TQ1` as the integration key and `Zh5eGmUq9zpfQnyUIu5OL9iWoMMv5ZNmk3zLJ4Ep` as the secret key:

```
Date: Tue, 21 Aug 2012 17:29:18 -0000
Authorization: Basic RElXSjhYNkFFWU9SNU9NQzZUUTE6YzFlZjQzNzY3YzNlYjNiMzI1OGRiZGRjYTZmOGQwOTQxZTA4
Host: api-XXXXXXXX.duosecurity.com
Content-Length: 35
Content-Type: application/x-www-form-urlencoded
```

Separate HTTP request header lines with CRLF newlines.

The following Python function can be used to construct the "Authorization" and "Date" headers:

Python 3 Python 2

```python
import base64, email.utils, hmac, hashlib, urllib

def sign(method, host, path, params, skey, ikey):
```

```
    """
    Return HTTP Basic Authentication ("Authorization" and "Date") headers.
    method, host, path: strings from request
    params: dict of request parameters
    skey: secret key
    ikey: integration key
    """

    # create canonical string
    now = email.utils.formatdate()
    canon = [now, method.upper(), host.lower(), path]
    args = []
    for key in sorted(params.keys()):
        val = params[key].encode("utf-8")
        args.append(
            '%s=%s' % (urllib.parse.
                       quote(key, '~'), urllib.parse.quote(val, '~')))
    canon.append('&'.join(args))
    canon = '\n'.join(canon)

    # sign canonical string
    sig = hmac.new(bytes(skey, encoding='utf-8'),
                   bytes(canon, encoding='utf-8'),
                   hashlib.sha1)
    auth = '%s:%s' % (ikey, sig.hexdigest())

    # return headers
    return {'Date': now, 'Authorization': 'Basic %s' % base64.b64encode(bytes(auth, encoding
```

If you receive 401 error responses to your API requests, check the following:

- Is the `Authorization` header correctly formatted? If not, you may receive a 40101 error.
- Does your framework override the `Date` header? The HTTP `Date:` header must be exactly the same string as was signed. This could result in a 40103 error.
- Are the `Date` and time zone used RFC 3339 compliant?? If not, you may get a 40104 or 40105 response.
- Are the parameters lexicographically sorted?
- Did you include a line for parameters when constructing the signature, even if you're not passing in any parameters?
- Are any hex digits lower-case?
- Are the `Content-Length` and `Content-Type` parameters correct? If not, your parameters may be ignored or you may receive a 40103 response because your signature considered parameters that the service didn't receive.

Users

Retrieve Users

Returns a paged list of users. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. If `username` is not provided, the list will contain all users. If `username` is provided, the list will either contain a single user (if a match was found) or no users. Requires "Grant resource - Read" API permission.

`GET` `/admin/v1/users`

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description

`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `300`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`
`usernames`	Deprecated	Retrieve specific users by specifying up to 100 `usernames` values. List format: `usernames=cjones&usernames=mwong&...etc`. Ignores other paging parameters when used.
`user_ids`	Deprecated	Retrieve specific users by specifying up to 100 `user_ids` values. List format: `user_ids=DUAAAAAAAAAAAAAAAAA&user_ids=DUBBBBBBBBBBBBBBBBB&...etc`. Ignores other paging parameters when used.
`user_id_list`	Optional	A list of user ids used to fetch multiple users by `user_id`. You can provide up to 100 `user_id` values. If you provide this parameter, you must not provide the `username`, `email` or `user_name_list` parameters. The `limit` and `offset` parameters will be ignored. Must be a JSON serialized array.
`username_list`	Optional	A list of usernames used to fetch multiple users by `username`. You can provide up to 100 `usernames`. If you provide this parameter, you must not provide the `username`, `email` or `user_name_list` parameters. The `limit` and `offset` parameters will be ignored. Must be a JSON serialized array.

Parameter	Required?	Description
`username`	Optional	Specify a user name (or username alias) to look up a single user.
`email`	Optional	Specify an email address to look up a single user.

RESPONSE CODES

Response	Meaning
200	Success. Returns a list of users.
400	Invalid parameters.

RESPONSE FORMAT

Key	Value
`alias1...4`	The user's username alias(es). Values included for backwards compatibility and reflect the same information as `aliases`.
`aliases`	Map of the user's username alias(es). Up to eight aliases may exist.

`created`	The user's creation date as a Unix timestamp.
`email`	The user's email address.
`enable_auto_prompt`	If `true`, the user is automatically prompted to use their last-used authentication method when authenticating. If `false`, the user is shown a list of authentication methods to initiate authentication. Only effective in the Universal Prompt.
`external_id`	The user's unique identifier imported by a directory sync. This is the `id` if the user is synced from Entra ID or `object_guid` if the user is synced from Active Directory. Not returned for users managed by OpenLDAP sync or users not managed by a directory sync.
`firstname`	Legacy parameter; returns no value. The user's given name.
`groups`	List of groups to which this user belongs. See Retrieve Groups for response info.
`is_enrolled`	Is `true` if the user has a phone, hardware token, U2F token, WebAuthn security key, or other WebAuthn method available for authentication. Otherwise, `false`.
`last_directory_sync`	An integer indicating the last update to the user via directory sync as a Unix timestamp, or `null` if the user has never synced with an external directory or if the directory that originally created the user has been deleted from Duo.
`last_login`	An integer indicating the last time this user logged in, as a Unix timestamp, or `null` if the user has not logged in.
`lastname`	Legacy parameter; returns no value. The user's surname.
`lockout_reason`	The user's lockout_reason. One of: **Reason** / **Description** "Failed Attempts" — The user was locked out due to excessive authentication attempts. "Not enrolled" — The user was locked out due to being not enrolled for a given period of time after the user was created. "Admin disabled" — The user was locked out by an admin from Duo Trust Monitor. "Admin API disabled" — The user's status was set to "locked out" by Admin API.
`notes`	Notes about this user. Viewable in the Duo Admin Panel.
`phones`	A list of phones that this user can use. See Retrieve Phones for descriptions of the phone response values.
`realname`	The user's real name (or full name).
`status`	The user's status. One of: **Status** / **Description** "active" — The user must complete secondary authentication. "bypass" — The user will bypass secondary authentication after completing primary authentication. "disabled" — The user will not be able to log in.

"locked out"	The user has been locked out due to a specific reason stored in the “lockout_reason” field.
"pending deletion"	The user was marked for deletion by a Duo admin from the Admin Panel, by the system for inactivity, or by directory sync. If not restored within seven days the user is permanently deleted.

Note that when a user is a member of a group, the group status may override the individual user's status. Group status is not shown in the user response.

`tokens` — A list of tokens that this user can use. See Retrieve Hardware Tokens for descriptions of the response values.

`u2f_tokens` — A list of U2F tokens that this user can use. U2F tokens were deprecated in Duo in February 2022.

Key	Value
`date_added`	The date the U2F token was registered in Duo.
`registration_id`	The U2F token's registration identifier. Use with GET token by ID.

`user_id` — The user's ID.

`username` — The user's username.

`webauthncredentials` — A list of WebAuthn authenticators that this user can use. See Retrieve WebAuthn Credentials by User ID for descriptions of the response values.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": [
    {
      "alias1": "joe.smith",
      "alias2": "jsmith@example.com",
      "alias3": null,
      "alias4": null,
      "aliases": {
        "alias1": "joe.smith",
        "alias2": "jsmith@example.com"
      },
      "created": 1489612729,
      "email": "jsmith@example.com",
      "enable_auto_prompt": true,
      "external_id": "1a2345b6-7cd8-9e0f-g1hi-23j45kl6m789",
      "firstname": "",
      "groups": [
        {
        "desc": "People with hardware tokens",
        "group_id": "DGBDKSSH37KSJ373JKSU",
        "mobile_otp_enabled": false,
        "name": "token_users",
        "push_enabled": false,
        "sms_enabled": false,
        "status": "Active",
        "voice_enabled": false
        }
      ],
```

```
    "is_enrolled": true,
    "last_directory_sync": 1508789163,
    "last_login": 1343921403,
    "lastname": "",
    "lockout_reason": null,
    "notes": "",
    "phones": [
      {
      "activated": true,
      "capabilities": [
        "auto",
        "push",
        "sms",
        "phone",
        "mobile_otp"
      ],
      "encrypted": "Encrypted",
      "extension": "",
      "fingerprint": "Configured",
      "last_seen": "2019-11-18T15:51:13",
      "model": "Apple iPhone 11 Pro",
      "name": "My iPhone",
      "number": "15555550100",
      "phone_id": "DPFZRS9FB0D46QFTM899",
      "platform": "Apple iOS",
      "postdelay": "0",
      "predelay": "0",
      "screenlock": "Locked",
      "sms_passcodes_sent": true,
      "tampered": "Not tampered",
      "type": "Mobile"
      }
    ],
    "realname": "Joe Smith",
    "status": "active",
    "tokens": [
      {
      "serial": "123456",
      "token_id": "DHIZ34ALBA2445ND4AI2",
      "type": "d1"
      }
    ],
    "u2ftokens": [],
    "user_id": "DU3RP9I2WOC59VZX672N",
    "username": "jsmith",
    "webauthncredentials": [
      {
      "credential_name": "Touch ID",
      "date_added": 1550685154,
      "label": "Touch ID",
      "webauthnkey": "WABFEOE007ZMV1QAZTRB"
      },
      {
      "credential_name": "YubiKey C",
      "date_added": 1550674764,
      "label": "Security Key",
      "webauthnkey": "WA4BD9AUVMSNUFWZGES4"
      }
    ],
  },
  {
    "alias1": "chris.jones",
```

```
    "alias2": "cjones@example.com",
    "alias3": null,
    "alias4": null,
    "aliases": {
      "alias1": "chris.jones",
      "alias2": "cjones@example.com"
    },
    "created": 1489612829,
    "email": "cjones@example.com",
    "enable_auto_prompt": false,
    "firstname": "",
    "groups": [],
    "is_enrolled": true,
    "last_directory_sync": null,
    "last_login": 1343821403,
    "lastname": "",
    "notes": "",
    "phones": [
      {
      "activated": true,
      "capabilities": [
        "auto",
        "push",
        "sms",
        "phone",
        "mobile_otp"
      ],
      "encrypted": "Encrypted",
      "extension": "",
      "fingerprint": "Configured",
      "last_seen": "2019-11-19T15:51:13",
      "model": "Google Pixel 3",
      "name": "Pixel3",
      "number": "15555550200",
      "phone_id": "DPFZRS9FB0D46QFTP00L",
      "platform": "Google Android",
      "postdelay": "0",
      "predelay": "0",
      "screenlock": "Locked",
      "sms_passcodes_sent": false,
      "tampered": "Not tampered",
      "type": "Mobile"
      }
    ],
    "realname": "Chris Jones",
    "status": "active",
    "tokens": [],
    "u2ftokens": [],
    "user_id": "DU3RP9I2WOC59VZXJ05H",
    "username": "cjones",
    "webauthncredentials": [
      {
      "credential_name": "YubiKey",
      "date_added": 1550564764,
      "label": "yubikey",
      "webauthnkey": "WA4BD9AUVMSNUFWZJ05H"
      }
    ]
  }
 ]
}
```

Create User

Create a new user with the specified `username`. Requires "Grant resource - Write" API permission.

```
POST  /admin/v1/users
```

PARAMETERS

Parameter	Required?	Description
`username`	**Required**	The name of the user to create.
`alias1...4`	Optional	A username alias for the user. Up to four aliases may be specified with this parameter. Aliases must be unique amongst users. This parameter maintained for backwards compatibility. Mutually exclusive with `aliases`.
`aliases`	Optional	Username aliases for the user. Up to eight aliases may be specified with this parameter as a set of URL-encoded key-value pairs e.g. `alias1=joe.smith&alias2=jsmith@example.com`. Ignores alias position values not specified. Aliases must be unique amongst users. Mutually exclusive with `alias1...4`.
`realname`	Optional	The real name (or full name) of this user.
`email`	Optional	The email address of this user.
`enable_auto_prompt`	Optional	If set to `0`, the user will be shown a list of authentication methods to initiate authentication. If set to `1`, the user will be automatically prompted to use their last-used authentication method. Only effective in the Universal Prompt. Default: `1`.
`status`	Optional	The user's status. One of: **Status** — **Description** "active" — The user must complete secondary authentication. This is the default value if no status is specified. "bypass" — The user will bypass secondary authentication after completing primary authentication. "disabled" — The user will not be able to complete secondary authentication.
`notes`	Optional	An optional description or notes field. Can be viewed in the Duo Admin Panel.
`firstname`	Optional	Legacy parameter; no effect if specified. The user's given name.
`lastname`	Optional	Legacy parameter; no effect if specified. The user's surname.

RESPONSE CODES

Response	Meaning
200	Success. Returns the newly created user.
400	Invalid or missing parameters, or user already exists with the given `username`.

RESPONSE FORMAT

Returns the new single user object. Refer to Retrieve Users for an explanation of the object's keys.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response":{
    "alias1": null,
    "alias2": null,
    "alias3": null,
    "alias4": null,
    "aliases": {},
    "created": 1657222760,
    "email": "jperez@example.com",
    "enable_auto_prompt": true,
    "firstname": "",
    "groups": [],
    "is_enrolled": false,
    "last_directory_sync": null,
    "last_login": null,
    "lastname": "",
    "lockout_reason": null,
    "notes": "",
    "phones": [],
    "realname": "Juan Perez",
    "status": "active",
    "tokens": [],
    "u2ftokens": [],
    "user_id": "DU0W79YFWZAJWJV6P00L",
    "username": "jperez",
    "webauthncredentials": []
  }
}
```


Create Multiple Users

Create multiple users at once. If one user fails to add, the entire operation fails. You can create a maximum of 100 users per request at a rate limit of 50 calls per minute. Requires "Grant resource - Write" API permission.

`POST` `/admin/v1/users/bulk_create`

PARAMETERS

Parameter	Required?	Description
`users`	**Yes**	Must be a serialized JSON list of objects, each of which contains information for a new user. For more information, see the list of user object attributes.

USER OBJECT ATTRIBUTES

Parameter	Required?	Description
`username`	**Yes**	Username of the user to create.
`realname`	No	The real name (or full name) of this user.
`email`	No	The email address of this user.
`status`	No	The user's status. One of: **Status** / **Description** "active" — The user must complete secondary authentication. This is the default value if no status is specified.

		"bypass"	The user will bypass secondary authentication after completing primary authentication.
		"disabled"	The user will not be able to complete secondary authentication.
`notes`	No	An optional description or notes field. Can be viewed in the Duo Admin Panel.	
`firstname`	No	Legacy parameter; no effect if specified. The user's given name.	
`lastname`	No	Legacy parameter; no effect if specified. The user's surname.	

RESPONSE FORMAT

Returns a list of response objects, one for each operation provided. The response format for each operation will follow the same format as that of the corresponding single-operation endpoint for that operation. For example, a Create User operation will receive a response in the format described in Create User.

Note that it is possible for some operations to fail while others succeed. The list of responses can include both successful responses as well as unsuccessful responses due to invalid parameters or rate limiting.

RESPONSE CODES

Response	Meaning
200	Success. Returns the list of newly created users.
400	Invalid or missing parameters, or user already exists for the given `username` provided.
429	Rate limited.

EXAMPLE REQUEST

Here's a sample python script that uses the admin api client to call out to the new endpoint. Note that the "users" param is a JSON serialized string.

```
{
#!/usr/bin/python
from __future__ import absolute_import
from __future__ import print_function
import pprint
import sys
import json

import duo_client
from six.moves import input

# Configuration and information about objects to create.
admin_api = duo_client.Admin(
    ikey='DIWJ8X6AEYOR5OMC6TQ1',
    skey='Zh5eGmUq9zpfQnyUIu5OL9iWoMMv5ZNmk3zLJ4Ep',
    host='api-XXXXXXXX.duosecurity.com',
    ca_certs='DISABLE'
)

response = admin_api.json_api_call(
    'POST',
    '/admin/v1/users/bulk_create',
    {
        'users': json.dumps([
            {
                'username': 'example_username_1',
```

```
                'email': 'example_user_1@example.com'
            },
            {
                'username': 'example_username_2',
                'status': 'disabled'
            }
        ])
    }
)

pprint.pprint(response)
}
```

EXAMPLE RESPONSE

Here's the output from running the above script. Note that the shape of the user objects returned here is consistent with the shape returned from other Admin API user operations (e.g. creating a single user). The returned users will be in the same order that they were received.

```
{
  "[{'alias1': None,
  'alias2': None,
  'alias3': None,
  'alias4': None,
  'aliases': {},
  'created': 1677770740,
  'email': 'example_user_1@example.com',
  'firstname': None,
  'groups': [],
  'is_enrolled': False,
  'last_directory_sync': None,
  'last_login': None,
  'lastname': None,
  'lockout_reason': "null",
  'notes': '',
  'phones': [],
  'realname': '',
  'status': 'active',
  'tokens': [],
  'u2ftokens': [],
  'user_id': 'DUVI0UT8SKB5P70WG42Z',
  'username': 'example_username_1',
  'webauthncredentials': []},
 {'alias1': None,
  'alias2': None,
  'alias3': None,
  'alias4': None,
  'aliases': {},
  'created': 1677770740,
  'email': '',
  'firstname': None,
  'groups': [],
  'is_enrolled': False,
  'last_directory_sync': None,
  'last_login': None,
  'lastname': None,
  'notes': '',
  'phones': [],
  'realname': '',
  'status': 'disabled',
  'tokens': [],
  'u2ftokens': [],
```

```
  'user_id': 'DU2X1VOO31O10F2ZNF43',
  'username': 'example_username_2',
  'webauthncredentials': []}]
  }
}
```

Retrieve User by ID

Return the single user with `user_id`. Requires "Grant resource - Read" API permission.

```
GET  /admin/v1/users/[user_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No user was found with the given `user_id`.

RESPONSE FORMAT

Returns a single user object. Refer to Retrieve Users for an explanation of the object's keys.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": {
    "alias1": "joe.smith",
    "alias2": "jsmith@example.com",
    "alias3": null,
    "alias4": null,
    "aliases": {
      "alias1": "joe.smith",
      "alias2": "jsmith@example.com"
    },
    "created": 1489612729,
    "email": "jsmith@example.com",
    "enable_auto_prompt": true,
    "external_id": "1a2345b6-7cd8-9e0f-g1hi-23j45kl6m789",
    "firstname": "",
    "groups": [
      {
      "desc": "People with hardware tokens",
      "group_id": "DGBDKSSH37KSJ373JKSU",
      "mobile_otp_enabled": false,
      "name": "token_users",
      "push_enabled": false,
      "sms_enabled": false,
      "status": "Active",
      "voice_enabled": false
      }
    ],
    "is_enrolled": true,
    "last_directory_sync": 1508789163,
    "last_login": 1343921403,
    "lastname": "",
```

```
    "lockout_reason": null,
    "notes": "",
    "phones": [
      {
      "activated": true,
      "capabilities": [
        "auto",
        "push",
        "sms",
        "phone",
        "mobile_otp"
        ],
      "encrypted": "Encrypted",
      "extension": "",
      "fingerprint": "Configured",
      "last_seen": "2019-11-18T15:51:13",
      "model": "Apple iPhone 11 Pro",
      "name": "My iPhone",
      "number": "15555550100",
      "phone_id": "DPFZRS9FB0D46QFTM899",
      "platform": "Apple iOS",
      "postdelay": "0",
      "predelay": "0",
      "screenlock": "Locked",
      "sms_passcodes_sent": true,
      "tampered": "Not tampered",
      "type": "Mobile"
      }
    ],
    "realname": "Joe Smith",
    "status": "active",
    "tokens": [
      {
      "serial": "123456",
      "token_id": "DHIZ34ALBA2445ND4AI2",
      "type": "d1"
      }
    ],
    "u2ftokens": [],
    "user_id": "DU3RP9I2WOC59VZX672N",
    "username": "jsmith",
    "webauthncredentials": [
      {
      "credential_name": "Touch ID",
      "date_added": 1550685154,
      "label": "Touch ID",
      "webauthnkey": "WABFEOE007ZMV1QAZTRB"
      },
      {
      "credential_name": "YubiKey C",
      "date_added": 1550674764,
      "label": "Security Key",
      "webauthnkey": "WA4BD9AUVMSNUFWZGES4"
      }
    ],
  }
}
```


Modify User

Change the username, username aliases, full name, status, and/or notes section of the user with ID `user_id`. Requires "Grant resource - Write" API permission.

`POST` `/admin/v1/users/[user_id]`

PARAMETERS

Parameter	Required?	Description
`username`	Optional	The new username.
`alias1...4`	Optional	A username alias for the user. Up to four aliases may be specified with this parameter. Aliases must be unique amongst users. This parameter maintained for backwards compatibility. Mutually exclusive with `aliases`.
`aliases`	Optional	Username aliases for the user. Up to eight aliases may be specified with this parameter as a set of URL-encoded key-value pairs e.g. `alias1=joe.smith&alias2=jsmith@example.com`. Ignores alias position values not specified. Remove the value for an existing alias by specifying a blank value e.g. `alias1=`. Aliases must be unique amongst users. Mutually exclusive with `alias1...4`.
`realname`	Optional	The new real name (or full name).
`email`	Optional	The new email address.
`enable_auto_prompt`	Optional	If set to `0`, the user will be shown a list of authentication methods to initiate authentication. If set to `1`, the user will be automatically prompted to use their last-used authentication method. Only effective in the Universal Prompt. Default: `1`.
`status`	Optional	The new status. Must be one of "active", "disabled", or "bypass", or "locked out". See Retrieve Users for an explanation of these fields. The "disabled" status may not be set via Admin API for users managed by Active Directory or Entra ID sync, nor can the API set "active" status for users disabled by directory sync.
`notes`	Optional	The new notes field.
`firstname`	Optional	Legacy parameter; no effect if specified. The user's new given name.
`lastname`	Optional	Legacy parameter; no effect if specified. The user's new surname.

RESPONSE CODES

Response	Meaning
200	The user was modified successfully. The user object is also returned (see Retrieve Users).
400	Invalid or missing parameters.
404	No user was found with the given `user_id`, or user already exists with the given `username`.

RESPONSE FORMAT

Returns the modified single user object. Refer to Retrieve Users for an explanation of the object's keys.

EXAMPLE RESPONSE

Same as Retrieve User by ID.

Delete User

Delete the user with ID `user_id` from the system. The API will not delete phones associated only with that user right away; remove them immediately with Delete Phone. This method returns 200 if the phone was found or if no such phone exists. Requires "Grant resource - Write" API permission.

Users deleted by the API do not get moved into the Trash view as "Pending Deletion" as they would if removed by directory sync, user deletion, or interactively from the Duo Admin Panel, and therefore are not available for restoration. Users deleted via the API are *immediately and permanently* removed from Duo.

```
DELETE  /admin/v1/users/[user_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The user was deleted or did not exist.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Enroll User

Enroll a user with user name `username` and email address `email` and send them an enrollment email that expires after `valid_secs` seconds. Requires "Grant resource - Write" API permission.

```
POST  /admin/v1/users/enroll
```

PARAMETERS

Parameter	Required?	Description
`username`	**Required**	The user name (or username alias) of the user to enroll.
`email`	**Required**	The email address of this user.
`valid_secs`	Optional	The number of seconds the enrollment code should remain valid. Default: `2592000` (30 days).

RESPONSE CODES

Response	Meaning
200	The enrollment code was generated and the user was sent an enrollment email. The newly created enrollment code is also returned.
400	Invalid or missing parameter(s), or the user with the given `username` and `email` address already exists and is enrolled.

RESPONSE FORMAT

Single string (the enrollment code).

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": "00d70e730b22cb66"
}
```

Create Bypass Codes for User

Clear all existing bypass codes for the user with ID `user_id` and return a list of `count` newly generated bypass codes, or specify `codes` that expire after `valid_secs` seconds, or `reuse_count` uses. To preserve existing bypass codes instead of clearing them the request must specify `preserve_existing=true`.

Requires "Grant resource - Write" API permission.

Object limits: 100 bypass codes per user.

`POST` `/admin/v1/users/[user_id]/bypass_codes`

PARAMETERS

Parameter	Required?	Description
`count`	Optional	Number of new bypass codes to create. At most 10 codes (the default) can be created at a time. Codes will be generated randomly.
`codes`	Optional	CSV string of codes to use. Mutually exclusive with count.
`preserve_existing`	Optional	Preserves existing bypass codes while creating new ones. Either `true` or `false`; effectively `false` if not specified. If true and the request would result the target user reaching the limit of 100 codes per user, or if `codes` is used and specifies a bypass code that already exists for the target user, then an error is returned and no bypass codes are created for nor cleared from the user.
`reuse_count`	Optional	The number of times generated bypass codes can be used. If `0`, the codes will have an infinite reuse_count. Default: `1`.
`valid_secs`	Optional	The number of seconds for which generated bypass codes remain valid. If `0` (the default) the codes will never expire.

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid or missing parameters, or one-to-many object limit reached.
404	No user was found with the given `user_id`.
500	Other internal error.

RESPONSE FORMAT

List of strings.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": [
    "407176182",
    "016931781",
    "338390347",
```

```
    "537828175",
    "006165274",
    "438680449",
    "877647224",
    "196167433",
    "719424708",
    "727559878"
  ]
}
```

Retrieve Bypass Codes by User ID

Returns a paged list of bypass code metadata associated with the user with ID `user_id`. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. Does not return the actual bypass codes. Requires "Grant resource - Read" API permission.

`GET` `/admin/v1/users/[user_id]/bypass_codes`

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No user was found with the given `user_id`.

RESPONSE FORMAT

Key	Value
`admin_email`	The email address of the Duo administrator who created the bypass code.
`bypass_code_id`	The bypass code's identifier. Use with GET bypass code by ID.
`created`	The bypass code creation date timestamp.
`expiration`	An integer indicating the expiration timestamp of the bypass code, or `null` if the bypass code does not expire on a certain date.
`reuse_count`	An integer indicating the number of times the bypass code may be used before expiring, or `null` if the bypass code has no limit on the number of times it may be used.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": [
    {
    "admin_email": "janesmith@example.com",
    "bypass_code_id": "DB2A9F0012RL54001FA3",
    "created": 1522260759,
    "expiration": 1522264359,
    "reuse_count": 1
    }
  ]
}
```


Retrieve Groups by User ID

Returns a paged list of groups associated with the user with ID `user_id`. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. Requires "Grant resource - Read" API permission.

```
GET  /admin/v1/users/[user_id]/groups
```

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No user was found with the given `user_id`.

RESPONSE FORMAT

Returns the groups for the user object. Refer to Retrieve Groups for an explanation of the object keys.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": [
    {
    "desc": "This is group A",
    "group_id": "DGBDKSSH37KSJ373JKSU",
    "mobile_otp_enabled": false,
```

```
    "name": "Group A",
    "push_enabled": false,
    "sms_enabled": false,
    "status": "active",
    "voice_enabled": false
    },
    {
    "desc": "This is group B",
    "group_id": "DGJKSLSH393YSJD93HSD3",
    "mobile_otp_enabled": false,
    "name": "Group B (from Microsoft Entra ID sync \"Acme Corp Entra ID\")",
    "push_enabled": false,
    "sms_enabled": false,
    "status": "active",
    "voice_enabled": false
    }
  ]
}
```


Associate Group with User

Associate a group with ID `group_id` with the user with ID `user_id`. Requires "Grant resource - Write" API permission.

Object limits: 100 groups per user.

`POST` `/admin/v1/users/[user_id]/groups`

PARAMETERS

Parameter	Required?	Description
`group_id`	**Required**	The ID of the group to associate with the user.

RESPONSE CODES

Response	Meaning
200	Success. Returns a response of "".
400	Invalid or missing parameters, one-to-many object limit reached, or nonexistent `group_id`. Also returns "Operation Failed" if the group does not exist.
404	Nonexistent `user id`.
500	Other internal error.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": ""
}
```


Disassociate Group from User

Disassociate a group from the user with ID `user_id`. This method will return 200 if the group was found or if no such group exists. Requires "Grant resource - Write" API permission.

`DELETE` `/admin/v1/users/[user_id]/groups/[group_id]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success, or no such group exists.
404	No user was found with the given `user_id`.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Retrieve Phones by User ID

Returns a paged list of phones associated with the user with ID `user_id`. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. Requires "Grant resource - Read" API permission.

```
GET  /admin/v1/users/[user_id]/phones
```

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No user was found with the given `user_id`.

RESPONSE FORMAT

Same as for Retrieve Phones, except phones have no `users` attribute.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": [
    {
    "activated": false,
    "capabilities": [
      "sms",
      "phone",
      "push"
    ],
    "encrypted": "Encrypted",
    "extension": "",
    "fingerprint": "Configured",
    "last_seen": "2019-03-04T15:04:04",
    "model": "Google Pixel 2 Xl",
    "name": "",
    "number": "+15035550102",
    "phone_id": "DPFZRS9FB0D46QFTM890",
    "platform": "Google Android",
    "postdelay": "",
    "predelay": "",
    "screenlock": "Locked",
    "sms_passcodes_sent": true,
    "tampered": "Not tampered",
    "type": "Mobile"
    },
    {
    "activated": false,
    "capabilities": [
      "phone"
    ].
    "encrypted": "",
    "extension": "",
    "fingerprint": "",
    "last_seen": "",
    "model": "Unknown",
    "name": "",
    "number": "+15035550103",
    "phone_id": "DPFZRS9FB0D46QFTM891",
    "platform": "Unknown",
    "postdelay": "",
    "predelay": "",
    "screenlock": "",
    "sms_passcodes_sent": false,
    "tampered": "",
    "type": "Landline"
    }
  ]
}
```


Associate Phone with User

Associate a phone with the user with ID `user_id`. Requires "Grant resource - Write" API permission.

Object limits: 100 phones per user; 100 users per phone.

`POST` `/admin/v1/users/[user_id]/phones`

PARAMETERS

Parameter	Required?	Description
`phone_id`	**Required**	The ID of the phone to associate with the user.

RESPONSE CODES

Response	Meaning
200	Success. Returns a response of "".
400	Invalid or missing parameters, one-to-many object limit reached, or nonexistent `phone_id`.
404	Nonexistent `user_id`.
500	Other internal error.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Disassociate Phone from User

Disassociate a phone from the user with ID `user_id`. The API will not automatically delete the phone after removing the last user association; remove it permanently with Delete Phone. This method returns 200 if the phone was found or if no such phone exists. Requires "Grant resource - Write" API permission.

```
DELETE  /admin/v1/users/[user_id]/phones/[phone_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success, or no such phone exists.
404	No user was found with the given `user_id`.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Retrieve Hardware Tokens by User ID

Returns a paged list of OTP hardware tokens associated with the user with ID `user_id`. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. Requires "Grant resource - Read" API permission.

`GET` `/admin/v1/users/[user_id]/tokens`

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No user was found with the given `user_id`.

RESPONSE FORMAT

Same as for Retrieve Hardware Tokens, except hardware tokens have no `admins` or `users` attribute.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": [
    {
    "serial": "123456",
    "token_id": "DHEKH0JJIYC1LX3AZWO4",
    "totp_step": null,
    "type": "d1"
    },
    {
    "serial": "123457",
    "token_id": "DHUNT3ZVS3ACF8AEV2WG",
    "totp_step": null,
    "type": "d1"
    }
  ]
}
```


Associate Hardware Token with User

Associate a hardware token with the user with ID `user_id`. Requires "Grant resource - Write" API permission.

Object limits: 100 tokens per user.

`POST` `/admin/v1/users/[user_id]/tokens`

PARAMETERS

Parameter	Required?	Description
`token_id`	**Required**	The ID of the hardware token to associate with the user.

RESPONSE CODES

Response	Meaning
200	Success. Returns a response of "".
400	Invalid or missing parameters, one-to-many object limit reached, `token_id` already in use by a different user, or nonexistent `token_id`.
404	Nonexistent `user_id`.
500	Other internal error.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Disassociate Hardware Token from User

Disassociate a hardware token from the user with ID `user_id`. This method will return 200 if the hardware token was found or if no such hardware token exists. Requires "Grant resource - Write" API permission.

```
DELETE  /admin/v1/users/[user_id]/tokens/[token_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No user was found with the given `user_id`

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Retrieve U2F Tokens by User ID

The U2F Tokens by User ID API endpoint `/admin/v1/users/[user_id]/u2ftokens` is deprecated as of February 2022. This API no longer allows listing all U2F tokens for a user. Requests to this endpoint now fail with the following response:

```
{
  "stat": "FAIL",
  "code": 40301,
  "message": "Access forbidden",
}
```

Retrieve WebAuthn Credentials by User ID

Returns a list of WebAuthn credentials associated with the user with ID `user_id`. Requires "Grant resource - Read" API permission.

```
GET  /admin/v1/users/[user_id]/webauthncredentials
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No user was found with the given `user_id`.

RESPONSE FORMAT

Key	Value
`credential_name`	Free-form label for the WebAuthn credential.
`date_added`	The date the WebAuthn credential was registered in Duo.
`label`	Indicates the type of WebAuthn credential. Example: `Windows Hello` or `iCloud Keychain`.
`webauthnkey`	The WebAuthn credential's registration identifier.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": [
    {
    "credential_name": "YubiKey 5",
    "date_added": 1550674764,
    "label": "Security Key",
    "webauthnkey": "WA4ED9AUVMSWUF00KES4"
    }
  ],
}
```

Retrieve Desktop Authenticators by User ID

Returns a paged list of desktop authenticators associated with the user with ID `user_id`. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. Requires "Grant resource - Read" API permission.

```
GET  /admin/v1/users/[user_id]/desktopauthenticators
```

PARAMETERS

Paging Parameter	Required?	Description

`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No user was found with the given `user_id`.

RESPONSE FORMAT

Key	Value
`daid`	The authenticator's ID.
`dakey`	The authenticator's Duo-specific identifier.
`device_name`	The endpoint's hostname.
`duo_desktop_version`	The version of Duo Desktop installed on the endpoint.
`os_family`	The endpoint's operating system platform.

EXAMPLE RESPONSE


```
{
  "response": [
    {
      "daid": 365,
      "dakey": "DDABCDE1FGHIJ23KL45",
      "device_name": "DESKTOP-ABC1234",
      "duo_desktop_version": "6.12.0",
      "os_family": "Windows"
    }
  ],
  "stat": "OK"
}
```


Synchronize User from Directory

Initiate a sync to create, update, or mark for deletion the user specified by `username` against the directory specified by the `directory_key`. The `directory_key` for a directory can be found by navigating to **Users → Directory Sync** in the Duo Admin Panel, and then clicking on the configured directory. Learn more about syncing individual users from Active Directory, OpenLDAP, or Entra ID. Requires "Grant resource - Write" API permission.

`POST` `/admin/v1/users/directorysync/[directory_key]/syncuser`

PARAMETERS

Parameter	Required?	Description
`username`	Required	The user to update or create via directory sync. This should be the same as the value for the user's username attribute in the source directory as configured in the sync.

RESPONSE CODES

Response	Meaning
200	The user was synced successfully and updated or added in Duo. The user object is also returned (see Retrieve Users).
404	The specified `username` or `directory_key` was incorrect, the user is not managed by the specified directory, or the user is not a member of any source directory group specified in the sync configuration.
429	Too many requests; try again later.

RESPONSE FORMAT

Returns the single synced user object with an additional `message` stating the user synced successfully. Refer to Retrieve Users for an explanation of the object's keys.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response":{
    "message": "User wwu synced successfully.",
    "user": {
      "alias1": "wwu@example.com",
      "alias2": null,
      "alias3": null,
      "alias4": null,
      "aliases": {
        "alias1": "wwu@example.com"
      },
      "created": 1553271753,
      "email": "",
      "enable_auto_prompt": true,
      "external_id": "1a2345b6-7cd8-9e0f-g1hi-23j45kl6m789",
      "firstname": "",
      "groups": [
        {
        "desc": "",
        "group_id": "DGKAT4WCV306FOONUNIC",
        "mobile_otp_enabled": false,
        "name": "Duo Users (from AD sync \"Acme Sync")",
        "push_enabled": false,
        "sms_enabled": false,
        "status": "Active",
        "voice_enabled": false
        }
    ],
    "is_enrolled": true,
    "last_directory_sync": 1657227493,
    "last_login": 1553271773,
    "lastname": "",
    "lockout_reason": null,
    "notes": "",
    "phones": [],
    "realname": "Wang Wu",
    "status": "active",
    "tokens": [],
    "u2ftokens": [],
    "user_id": "DUYGKDZI47WEJIIXU6QI",
    "username": "wwu",
    "webauthncredentials": [
      {
      "credential_name": "YubiKey",
```

```
        "date_added": 1553271760,
        "label": "Security Key",
        "webauthnkey": "WAYRDI633R1219HM1BN7"
        }
      ]
    }
  }
}
```


Send Verification Push

Sends a verification Duo Push to the user with ID `user_id`. Verification pushes can also be sent from the Duo Admin Panel. Requires "Grant resource - Write" API permission.

`POST` `/admin/v1/users/[user_id]/send_verification_push`

PARAMETERS

Parameter	Required?	Description
`phone_id`	Required	The ID of the phone belonging to the user. This phone should be activated for push.

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid parameters or phone cannot receive pushes.
404	No user was found with the given `user_id` or no phone was found with the given `phone_id`.

RESPONSE FORMAT

Key	Value
`confirmation_code`	The Duo Push sent to the user contains this confirmation code.
`push_id`	The ID of the Duo Push sent.

EXAMPLE RESPONSE


```
{
  "response": {
    "confirmation_code": "123456",
    "push_id": "123abc45-6def-789g-h012-34567ijk8901"
  },
  "stat": "OK"
}
```


Retrieve Verification Push Response

Retrieves the verification push result for the user with ID `user_id`. Push response information will be available for 120 seconds after the push was sent, after which this endpoint will return a 404. If no success or failure response was returned by this endpoint during these 120 seconds, it can be assumed that the push has timed out. Requires "Grant resource - Read" API permission.

`GET` `/admin/v1/users/[user_id]/verification_push_response`

PARAMETERS

Parameter	Required?	Description

`push_id`	Required	The ID of the Duo Push sent.

RESPONSE CODES

Response	Meaning
200	Success.
404	Invalid or expired `push_id`. Note that the `push_id` expires after 120 seconds.

RESPONSE FORMAT

Key	Value
`push_id`	The ID of the Duo Push sent.
`result`	The result of the verification push sent. One of: ◦ `approve`: User approved the push. ◦ `deny`: User denied the push. ◦ `fraud`: User marked the push as fraud. ◦ `waiting`: User has not responded to the push yet.

EXAMPLE RESPONSE

```
{
  "response": {
    "push_id": "123abc45-6def-789g-h012-34567ijk8901"
    "result": "approve"
  },
  "stat": "OK"
}
```

Bulk Operations

Bulk User Operations

Performs a list of operations serially in the order provided. You can perform a maximum of 50 operations per request at a rate limit of 50 calls per minute. Requires "Grant resource - Write" API permission.

`POST` `/admin/v1/bulk`

PARAMETERS

Parameter	Required?	Description
`operations`	**Yes**	Must be a JSON list of objects, each of which specifies an operation.

OPERATIONS

Each operation must be a JSON object that specifies the following attributes.

Parameter	Required?	Description
`method`	**Yes**	HTTP method “POST” or “DELETE”.
`path`	**Yes**	HTTP path for the operation. See below for a list of enabled operations.
`body`	**Yes**	A JSON object containing the parameters for this operation.

VALID OPERATIONS

Operation	Method	Path	Body
Create User	`POST`	`/admin/v1/users`	See Parameters section from Create User.
Modify User	`POST`	`/admin/v1/users/[user_id]`	See Parameters section from Modify User.
Delete User	`DELETE`	`/admin/v1/users/[user_id]`	See Parameters section from Delete User.
Associate Group with User	`POST`	`/admin/v1/users/[user_id]/groups`	See Parameters section from Associate Group with User.
Disassociate Group from User	`POST`	`/admin/v1/users/[user_id]/groups/[group_id]`	See Parameters section from Disassociate Group from User.

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid request, possibly due to the shape of the `operations` attribute being invalid.
429	Too many requests.

RESPONSE FORMAT

Returns a list of response objects, one for each operation provided. The response format for each operation will follow the same format as that of the corresponding single-operation endpoint for that operation. For example, a Create User operation will receive a response in the format described in Create User.

Note that it is possible for some operations to fail while others succeed. The list of responses can include both successful responses as well as unsuccessful responses due to invalid parameters or rate limiting.

EXAMPLE OPERATIONS LIST

This sample list of operations provides an example of what should be passed as the `operations` parameter.

```
{
[
    {
        "method": "POST",
        "path": "/admin/v1/users",
        "body": {
            "username": "uname1",
            "alias1": "my_alias1",
            "alias2": "my_alias2",
            "alias3": "my_alias3",
            "alias4": "my_alias4",
            "email": "user@example.com",
            "status": "active",
            "notes": "This is a user"
        }
    },
    {
        "method": "POST",
        "path": "/admin/v1/users/DUJ424R8DJFJ05HASDFJ",
        "body": {
```

```
            "username": "uname2",
            "alias2": "updated_alias2",
            "email": "user2@example.com",
            "status": "active",
            "notes": "This is another user"
        }
    },
    {
        "method": "DELETE",
        "path": "/admin/v1/users/DUJ424R8DJFJ05HASDFJ",
        "body": {}
    },
    {
        "method": "POST",
        "path": "/admin/v1/users/DUJ424R8DJFJ05HASDFJ/groups",
        "body": {
            "group_id": "DGBDKSSH37KSJ373JKSU",
        }
    },
    {
        "method": "DELETE",
        "path": "/admin/v1/users/DUJ424R8DJFJ05HASDFJ/groups/DGBDKSSH37KSJ373JKSU",
        "body": {}
    }
]

}
```

EXAMPLE RESPONSE

```
{
[
    {
      "stat": "OK",
      "response":{
        "alias1": "my_alias1",
        "alias2": "my_alias2",
        "alias3": "my_alias3",
        "alias4": "my_alias4",
        "aliases": {
            "alias1": "my_alias1",
            "alias2": "my_alias2",
            "alias3": "my_alias3",
            "alias4": "my_alias4",
      },
      "created": 1657222760,
      "email": "user@example.com",
    "enable_auto_prompt": true,
      "firstname": "",
      "groups": [],
      "is_enrolled": false,
      "last_directory_sync": null,
      "last_login": null,
      "lastname": "",
    "lockout_reason": null,
      "notes": "This is a user",
      "phones": [],
      "realname": "",
      "status": "active",
      "tokens": [],
      "u2ftokens": [],
      "user_id": "DUJ424R8DJFJ05HASDFJ",
```

```
      "username": "uname1",
      "webauthncredentials": []
    }
  },
  {
    "stat": "OK",
    "response":{
      "alias1": null,
      "alias2": "my_alias2",
      "alias3": null,
      "alias4": null,
      "aliases": {
          "alias2": "updated_alias2",
      },
      "created": 1657223860,
      "email": "user2@example.com",
    "enable_auto_prompt": true,
      "firstname": "",
      "groups": [],
      "is_enrolled": false,
      "last_directory_sync": null,
      "last_login": null,
      "lastname": "",
    "lockout_reason": null,
      "notes": "This is another user",
      "phones": [],
      "realname": "",
      "status": "active",
      "tokens": [],
      "u2ftokens": [],
      "user_id": "DUJ424R8DP0OL5D4DAFJ",
      "username": "uname2",
      "webauthncredentials": []
    }
  },
  {
      "stat": "OK",
      "response": ""
  },
  {
      "stat": "OK",
      "response": ""
    },
    {
        "stat": "OK",
        "response": ""
    }
]

}
```


Groups

Per-user Group Operations

See Retrieve Groups by User ID, Associate Group with User, and Disassociate Group from User.

Retrieve Groups

Returns a paged list of groups. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. Requires "Grant resource - Read" API permission.

`GET` `/admin/v1/groups`

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `100`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`
`group_ids`	Deprecated	Retrieve specific groups by specifying up to 200 `group_ids` values. List format: `group_ids=DGBDKSSH37KSJ373JKSU&group_ids=DGJKSLSH393YSJD93HSD3&...etc` Ignores other paging parameters when used.
`group_id_list`	Optional	A list of group ids used to fetch multiple groups by `group_ids`. You can provide up to 100 `group_ids`. If you provide this parameter, the `limit` and `offset` parameters will be ignored. Must be a JSON serialized array.

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	Success.

RESPONSE FORMAT

Key	Value
`desc`	The group's description.
`group_id`	The group's ID.
`mobile_otp_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`name`	The group's name. If managed by directory sync, then the name returned here also indicates the source directory.
`push_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`sms_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`status`	The group's authentication status. May be one of:

Status	Description
"Active"	The users in the group must complete secondary authentication.
"Bypass"	The users in the group will bypass secondary authentication after completing primary authentication.
"Disabled"	The users in the group will not be able to authenticate.

`voice_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": [
    {
    "desc": "This is group A",
    "group_id": "DGBDKSSH37KSJ373JKSU",
    "mobile_otp_enabled": false,
    "name": "Group A",
    "push_enabled": false,
    "sms_enabled": false,
    "status": "Active",
    "voice_enabled": false
    },
    {
    "desc": "This is group B",
    "group_id": "DGJKSLSH393YSJD93HSD3",
    "mobile_otp_enabled": false,
    "name": "Group B (from Microsoft Entra ID sync \"Acme Corp Entra ID\")",
    "push_enabled": false,
    "sms_enabled": false,
    "status": "Active",
    "voice_enabled": false
    }
  ],
}
```


Create Group

Create a new group. Requires "Grant resource - Write" API permission.

`POST` `/admin/v1/groups`

PARAMETERS

Parameter	Required?	Description
`name`	**Required**	The name of the group.
`desc`	Optional	The description of the group.
`push_enabled`	Optional	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`sms_enabled`	Optional	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.

`voice_enabled`	Optional	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`mobile_otp_enabled`	Optional	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`status`	Optional	The authentication status of the group. See Retrieve Groups for a list of possible values.

RESPONSE CODES

Response	Meaning
200	Success.
400	Group with given name already exists or one of the parameters is invalid.

RESPONSE FORMAT

Key	Value
`desc`	The group's description.
`push_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`sms_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`voice_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`mobile_otp_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`group_id`	The group's ID.
`name`	The group's name.
`status`	The group's authentication status. See Retrieve Groups for a list of possible values.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": {
    "desc": "This is an example group",
    "group_id": "DGBDKSSH37KSJ373JKSU",
    "mobile_otp_enabled": false,
    "name": "Example Group",
    "push_enabled": false,
    "sms_enabled": false,
    "status": "active",
    "voice_enabled": false
  }
}
```

Get Group Info

Retrieve information about a group. Note that this output does not include a list of group members. To retrieve group members, use `/admin/v2/groups/[group id]/users`. Requires "Grant resource - Read" API permission.

`GET` `/admin/v2/groups/[group_id]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	Group with given ID was not found.

RESPONSE FORMAT

Key	Value
`desc`	The group's description.
`push_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`sms_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`voice_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`mobile_otp_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`group_id`	The group's ID.
`name`	The group's name. If managed by directory sync, then the name returned here also indicates the source directory.
`status`	The group's authentication status. See Retrieve Groups for a list of possible values.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": {
    "desc": "Group description",
    "group_id": "DGBDKSSH37KSJ373JKSU",
    "mobile_otp_enabled": false,
    "name": "Group Name",
    "push_enabled": false,
    "sms_enabled": false,
    "status": "active",
    "voice_enabled": false
  }
}
```

Returns a paged list of members of a specified group. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value.

`GET` `/admin/v2/groups/[group_id]/users`

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid parameters.
404	Group with given ID was not found.

RESPONSE FORMAT

Key	Value
`user_id`	The user's ID.
`username`	The user's username.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "metadata": {
      "total_objects": 4
  },
  "response": [
      {
          "user_id": "DUJF3424R8DJFJ05HASD",
          "username": "user1"
      },
      {
          "user_id": "DUJSFP005D4DJFJ05HSD",
          "username": "user2"
      },
      {
          "user_id": "DUJSFP001D4DJFJB0OFS",
          "username": "user3"
      }
  ]
}
```

Get Group Info (Legacy v1)

The v1 groups endpoint limits the response to the first 4,000 group members. Consider migrating to the v2 endpoint.
Requires "Grant resource - Read" API permission.

```
GET /admin/v1/groups/[group_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	Group with given ID was not found.

RESPONSE FORMAT

Key	Value
`desc`	The group's description.
`push_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`sms_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`voice_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`mobile_otp_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`group_id`	The group's ID.
`name`	The group's name. If managed by directory sync, then the name returned here also indicates the source directory.
`status`	The group's authentication status. See Retrieve Groups for a list of possible values.
`users`	A list of the users in the group.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "desc": "Group description",
    "group_id": "DGBDKSSH37KSJ373JKSU",
    "mobile_otp_enabled": false,
    "name": "Group Name",
    "push_enabled": false,
    "sms_enabled": false,
    "status": "active",
    "users": [
      {
      "user_id": "DUJ424R8DP0OL5D4DAFJ",
      "username": "User A"
      },
      {
      "user_id": "DUJSP005D4DJFJ05HASD",
      "username": "User B"
      }
    ],
    "voice_enabled": false
  }
}
```

Update Group

Update information about a group. Requires "Grant resource - Write" API permission.

```
POST  /admin/v1/groups/[group_id]
```

PARAMETERS

Parameter	Required?	Description
name	Optional	Update the name of the group.
desc	Optional	Update the description of the group.
push_enabled	Optional	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
sms_enabled	Optional	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
voice_enabled	Optional	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
mobile_otp_enabled	Optional	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
status	Optional	The authentication status of the group. See Retrieve Groups for a list of possible values.

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid parameters.
404	Group with given ID was not found.

RESPONSE FORMAT

Key	Value
`desc`	The group's updated description.
`push_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`sms_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`voice_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`mobile_otp_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`group_id`	The group's ID.
`name`	The group's updated name.
`status`	The group's updated authentication status. See Retrieve Groups for a list of possible values.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "desc": "Group description",
    "group_id": "DGBDKSSH37KSJ373JKSU",
    "mobile_otp_enabled": false,
    "name": "Group Name",
    "push_enabled": false,
    "sms_enabled": false,
    "status": "active",
    "voice_enabled": false
  }
}
```


Delete Group

Delete a group. Requires "Grant resource - Write" API permission.

`DELETE` `/admin/v1/groups/[group_id]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The group was deleted or did not exist.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE


```
{
  "response": "",
  "stat": "OK"
}
```


Phones

Per-user Phone Operations

See Retrieve Phones by User ID, Associate Phone with User, and Disassociate Phone from User.

Retrieve Phones

Returns a paged list of phones. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. If no `number` or `extension` parameters are provided, the list will contain all phones. Otherwise, the list will contain either single phone (if a match was found), or no phones. Requires "Grant resource - Read" API permission.

`GET` `/admin/v1/phones`

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

Parameter	Required?	Description
`number`	Optional	Specify a phone number in E.164 format to look up a single phone.
`extension`	Optional	The extension, if necessary.

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid number.

RESPONSE FORMAT

Key	Value
`activated`	Has this phone been activated for Duo Mobile yet? Either `true` or `false`.
`capabilities`	List of strings, each a factor that can be used with the device. **Capability** / **Meaning** `auto` — The device is valid for automatic factor selection (e.g. phone or push). `push` — The device is activated for Duo Push. `phone` — The device can receive phone calls. `sms` — The device can receive batches of SMS passcodes. `mobile_otp` — The device can generate passcodes with Duo Mobile.
`encrypted`	The encryption status of an Android or iOS device file system. One of: "Encrypted", "Unencrypted", or "Unknown". Blank for other platforms. This information is available to Duo Premier and Duo Advantage plan customers.
`extension`	An extension, if necessary.
`fingerprint`	Whether an Android or iOS phone is configured for biometric verification. One of: "Configured", "Disabled", or "Unknown". Blank for other platforms. This information is available to Duo Premier and Duo Advantage plan customers.

`last_seen`	An integer indicating the timestamp of the last contact between Duo's service and the activated Duo Mobile app installed on the phone. Blank if the device has never activated Duo Mobile or if the platform does not support it.
`model`	The phone's model.
`name`	Free-form label for the phone.
`number`	The phone number in E.164 format. A phone with a smartphone platform but no number is a tablet.
`phone_id`	The phone's ID.
`platform`	The phone platform. One of: "unknown", "google android", "apple ios", "windows phone 7", "rim blackberry", "java j2me", "palm webos", "symbian os", "windows mobile", or "generic smartphone". "windows phone" is accepted as a synonym for "windows phone 7". This includes devices running Windows Phone 8. If a smartphone's exact platform is unknown but it will have Duo Mobile installed, use "generic smartphone" and generate an activation code. When the phone is activated its platform will be automatically detected.
`postdelay`	The time (in seconds) to wait after the extension is dialed and before the speaking the prompt.
`predelay`	The time (in seconds) to wait after the number picks up and before dialing the extension.
`screenlock`	Whether screen lock is enabled on an Android or iOS phone. One of: "Locked", "Unlocked", or "Unknown". Blank for other platforms. This information is available to Duo Premier and Duo Advantage plan customers.
`sms_passcodes_sent`	Have SMS passcodes been sent to this phone? Either `true` or `false`.
`tampered`	Whether an iOS or Android device is jailbroken or rooted. One of: "Not Tampered", "Tampered", or "Unknown". Blank for other platforms. This information is available to Duo Premier and Duo Advantage plan customers.
`type`	The type of phone. One of: "unknown", "mobile", or "landline".
`users`	A list of users associated with this phone. See Retrieve Users for descriptions of the response fields.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": [
    {
      "activated": true,
      "capabilities": [
        "auto",
        "push",
        "sms",
        "phone",
        "mobile_otp"
      ],
```

```
    "encrypted": "Encrypted",
    "extension": "",
    "fingerprint": "Configured",
    "last_seen": "2019-11-18T15:51:13",
    "model": "Apple iPhone 11 Pro",
    "name": "My iPhone",
    "number": "15555550100",
    "phone_id": "DPFZRS9FB0D46QFTM899",
    "platform": "Apple iOS",
    "postdelay": "",
    "predelay": "",
    "screenlock": "Locked",
    "sms_passcodes_sent": false,
    "tampered": "Not tampered",
    "type": "Mobile",
    "users": [
      {
      "alias1": "joe.smith",
      "alias2": "jsmith@example.com"
      "alias3": null,
      "alias4": null,
      "aliases": {
        "alias1": "joe.smith",
        "alias2": "jsmith@example.com"
      },
      "created": 1509717442,
      "email": "jsmith@example.com",
      "enable_auto_prompt": true,
      "firstname": "",
      "is_enrolled": false,
      "last_directory_sync": null,
      "last_login": 1474399627,
      "lastname": "",
      "notes": "",
      "realname": "Joe Smith",
      "status": "active",
      "user_id": "DUJZ2U4L80HT45MQ4EOQ",
      "username": "jsmith"
      }
    ]
  },
  {
    "activated": true,
    "capabilities": [
      "auto",
      "push",
      "sms",
      "phone",
      "mobile_otp"
    ],
    "encrypted": "Encrypted",
    "extension": "",
    "fingerprint": "Configured",
    "last_seen": "2019-11-19T15:51:13",
    "model": "Google Pixel 3",
    "name": "Pixel3",
    "number": "15555550200",
    "phone_id": "DPFZRS9FB0D46QFTP00L",
    "platform": "Google Android",
    "postdelay": "0",
    "predelay": "0",
    "screenlock": "Locked",
```

```
    "sms_passcodes_sent": false,
    "tampered": "Not tampered",
    "type": "Mobile"
    "users": [
      {
      "alias1": "chris.jones",
      "alias2": "cjones@example.com",
      "alias3": null,
      "alias4": null,
      "aliases": {
        "alias1": "chris.jones",
        "alias2": "cjones@example.com"
      },
      "created": 1489612829,
      "email": "cjones@example.com",
      "enable_auto_prompt": true,
      "firstname": "",
      "groups": [],
      "is_enrolled": true,
      "last_directory_sync": null,
      "last_login": 1343821403,
      "lastname": "",
      "notes": "",
      "realname": "Chris Jones",
      "status": "active",
      "user_id": "DU3RP9I2WOC59VZXJ05H",
      "username": "cjones"
      }
    ]
  }
 ]
}
```


Create Phone

Create a new phone with a specified phone number or other parameters. Requires "Grant resource - Write" API permission.

`POST` `/admin/v1/phones`

PARAMETERS

Parameter	Required?	Description
`number`	Optional	The phone number; E.164 format recommended (i.e. "+17345551212"). If no leading plus sign is provided then it is assumed to be a United States number and an implicit "+1" country code is prepended. Dashes and spaces are ignored. A phone with a smartphone platform but no number is a tablet.
`name`	Optional	Free-form label for the phone.
`extension`	Optional	The extension.
`type`	Optional	The phone type. See Retrieve Phones for a list of possible values.
`platform`	Optional	The phone platform. See Retrieve Phones for a list of possible values.
`predelay`	Optional	The time (in seconds) to wait after the number picks up and before dialing the extension.
`postdelay`	Optional	The time (in seconds) to wait after the extension is dialed and before the speaking the prompt.

RESPONSE CODES

Response	Meaning
200	The phone was created successfully. The newly created phone is also returned (see Retrieve Phones).
400	Invalid or missing parameter(s), or phone already exists with the given `number` and `extension`.

RESPONSE FORMAT

Returns the single phone object created. Refer to Retrieve Phones for an explanation of the object's keys.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response":{
    "activated": false,
    "capabilities": [],
    "extension": "",
    "last_seen": "",
    "model": "Unknown",
    "name": "",
    "number": "15555551111",
    "phone_id": "DP1D9EZJNZNQXJ05HKJB",
    "platform": "Generic Smartphone",
    "postdelay": "",
    "predelay": "",
    "sms_passcodes_sent": false,
    "type": "Unknown",
    "users": []
    }
}
```


Retrieve Phone by ID

Return the single phone with `phone_id`. Requires "Grant resource - Read" API permission.

```
GET  /admin/v1/phones/[phone_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No phone was found with the given `phone_id`.

RESPONSE FORMAT

Returns a single phone object. Refer to Retrieve Phones for an explanation of the object's keys.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "activated": true,
    "capabilities": [
      "auto",
```

```
      "push",
      "sms",
      "phone",
      "mobile_otp"
    ],
    "encrypted": "Encrypted",
    "extension": "",
    "fingerprint": "Configured",
    "last_seen": "2019-03-04T15:04:04",
    "model": "Google Pixel 2 Xl",
    "name": "",
    "number": "+15555550100",
    "phone_id": "DPFZRS9FB0D46QFTM899",
    "platform": "Google Android",
    "postdelay": "",
    "predelay": "",
    "screenlock": "Locked",
    "sms_passcodes_sent": false,
    "tampered": "Not tampered",
    "type": "Mobile",
    "users": [
      {
        "alias1": "joe.smith",
        "alias2": "jsmith@example.com"
        "alias3": null,
        "alias4": null,
        "aliases": {
          "alias1": "joe.smith",
          "alias2": "jsmith@example.com"
        },
        "created": 1509717442,
        "email": "jsmith@example.com",
        "enable_auto_prompt": true,
        "firstname": "",
        "is_enrolled": false,
        "last_directory_sync": null,
        "last_login": 1474399627,
        "lastname": "",
        "notes": "",
        "realname": "Joe Smith",
        "status": "active",
        "user_id": "DUJZ2U4L80HT45MQ4EOQ",
        "username": "jsmith",
      }
    ]
  }
}
```


Modify Phone

Change the details of the phone with ID `phone_id`. Requires "Grant resource - Write" API permission.

`POST` `/admin/v1/phones/[phone_id]`

PARAMETERS

Parameter	Required?	Description
`number`	Optional	The new phone number; E.164 format recommended (i.e. "+17345551212"). If no leading plus sign is provided then it is assumed to be a United States number and an implicit "+1" country code is prepended. Dashes and spaces are ignored.

`name`	Optional	Free-form label for the phone.
`extension`	Optional	The new extension.
`type`	Optional	The phone type. See Retrieve Phones for a list of possible values.
`platform`	Optional	The phone platform. See Retrieve Phones for a list of possible values.
`predelay`	Optional	The time (in seconds) to wait after the number picks up and before dialing the extension.
`postdelay`	Optional	The time (in seconds) to wait after the extension is dialed and before the speaking the prompt.

RESPONSE CODES

Response	Meaning
200	The phone was modified successfully. The phone object is returned.
400	Invalid or missing parameter(s), or phone already exists with the given `number` and `extension`.
404	No phone was found with the given `phone_id`.

RESPONSE FORMAT

Returns the modified single phone object. Refer to Retrieve Phones for an explanation of the object's keys.

EXAMPLE RESPONSE

Same as Retrieve Phone by ID.

Delete Phone

Delete the phone with ID `phone_id` from the system. Requires "Grant resource - Write" API permission.

`DELETE` `/admin/v1/phones/[phone_id]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The phone was deleted or did not exist.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Create Activation Code

Generate a Duo Mobile activation code. This method will fail if the phone's type or platform are Unknown. Requires "Grant resource - Write" API permission.

`POST` `/admin/v1/phones/[phone_id]/activation_url`

PARAMETERS

Parameter	Required?	Description
`valid_secs`	Optional	The number of seconds this activation code remains valid. Default: `86400` (one day). Expiration not supported for legacy phone platforms that support passcode generation only (not Duo Push).
`install`	Optional	Specify `1` to also return an installation URL for Duo Mobile; `0` to not return a URL. Default: `0`.

RESPONSE CODES

Response	Meaning
200	The activation code was successfully generated.
400	Invalid parameters or invalid phone. The phone's platform must be one on which Duo Mobile can be activated.
404	No phone was found with the given `phone_id`.

RESPONSE FORMAT

Key	Value
`activation_barcode`	URL of a QR code. Scan the code with Duo Mobile to complete activation. This QR code uses the same activation code as `activation_url`.
`activation_url`	Opening this URL on a phone with the Duo Mobile app installed will automatically complete activation.
`installation_url`	Opening this URL on the phone will prompt the user to install Duo Mobile. Only present if `install` was `1`.
`valid_secs`	An integer indicating the number of seconds that the activation code remains valid.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "activation_barcode": "https://api-abcdefgh.duosecurity.com/frame/qr?value=duo%3A%2F%2Factiva
    "activation_url": "https://m-abcdefgh.duosecurity.com/iphone/7dmi4Oowz5g3J47FARLs",
    "valid_secs": 3600
  }
}
```


Send Activation Code via SMS

Generate a Duo Mobile activation code and send it to the phone via SMS, optionally sending an additional message with a URL to install Duo Mobile. This method will fail if the phone's type or platform are Unknown. Requires "Grant resource - Write" API permission.

SMS Size Limits

The recommended maximum length for `activation_msg` and `installation_msg` is 80 characters.

Activation and installation SMS messages are limited to 160 characters or less. If providing custom text, please make sure to leave enough room for a URL to be sent in the same message. The exact length available for custom text varies depending on the device's platform and whether international characters were used. Activation URLs are typically about 60 characters long. Installation URLs are between 50 and 75 characters long.

`POST` `/admin/v1/phones/[phone_id]/send_sms_activation`

PARAMETERS

Parameter	Required?	Description
`valid_secs`	Optional	The number of seconds this activation code remains valid. Default: `86400` (one day).
`install`	Optional	Specify `1` to cause an installation SMS message to be sent before the activation message, or `0` to not send an installation SMS message. Default: `0`.
`installation_msg`	Optional	A custom installation message to send to the user. Only valid if installation was requested. Must contain the phrase "<insturl>", which is replaced with the installation URL.
`activation_msg`	Optional	A custom activation message to send to the user. Must contain "<acturl>", which is replaced with the activation URL.

RESPONSE CODES

Response	Meaning
200	The activation code was generated and sent successfully.
400	Invalid parameters or invalid phone. The phone must be able to receive SMS messages and its platform must be one on which Duo Mobile can be activated.
404	No phone was found with the given `phone_id`.
500	Failed to send SMS message or SMS message too long.

RESPONSE FORMAT

Key	Value
`activation_msg`	The text of the activation message.
`activation_barcode`	URL of a QR code. Scan the code with Duo Mobile to complete activation. This QR code contains the same activation code as `activation_url`.
`installation_msg`	The text of the installation message. Only present if the `install` parameter was set to `1` in the request.
`valid_secs`	An integer indicating the number of seconds that the activation URL remains valid.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "activation_barcode": "https://api-abcdef/frame/qr\?*value=duo%3A%2F%2FXoudqt8a9F-Jqt",
    "activation_msg": "To activate the Duo Mobile app, click this link: https://m-eval.duosecurit
    "installation_msg": "Welcome to Duo! To install the Duo Mobile app, click this link: http://m
    "valid_secs": 3600
  }
}
```


Send Installation URL via SMS

Send a message via SMS describing how to install Duo Mobile. This method will fail if the phone's type or platform are Unknown. Requires "Grant resource - Write" API permission.

SMS Size Limits

The recommended maximum length for `installation_msg` is 80 characters.

> Installation SMS messages are limited to 160 characters or less. If providing custom text, please make sure to leave enough room for a URL to be sent in the same message. The exact length available for custom text varies depending on the device's platform and whether international characters were used. Installation URLs are between 50 and 75 characters long.

`POST` `/admin/v1/phones/[phone_id]/send_sms_installation`

PARAMETERS

Parameter	Required?	Description
`installation_msg`	Optional	A custom installation message to send to the user. Must contain the phrase "<insturl>", which is replaced with the installation URL.

RESPONSE CODES

Response	Meaning
200	The installation URL was successfully sent.
400	Invalid parameters or invalid phone. The phone must be able to receive SMS messages and its platform must be one on which Duo Mobile can be activated.
404	No phone was found with the given `phone_id`.
500	Failed to send SMS message or SMS message too long.

RESPONSE FORMAT

Key	Value
`installation_msg`	The text of the installation message.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": {
    "installation_msg": "Welcome to Duo! To install the Duo Mobile app, click this link: http://m
  }
}
```

Send Passcodes via SMS

Generate a new batch of SMS passcodes send them to the phone in a single SMS message. Requires "Grant resource - Write" API permission.

`POST` `/admin/v1/phones/[phone_id]/send_sms_passcodes`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The passcodes were generated and sent successfully.
404	No phone was found with the given `phone_id`.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Tokens

Per-user Token Operations

See Retrieve Hardware Tokens by User ID, Associate Hardware Token with User, and Disassociate Hardware Token from User.

Per-administrator Token Operations

See Retrieve Administrator by ID, Create Administrator, and Modify Administrator. Note that token information retrieved from the Tokens endpoint does not include information about administrators associated with a token, just end-users.

Retrieve Hardware Tokens

Returns a paged list of OTP hardware tokens. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. If no `type` and `serial` parameters are provided, the list will contain all hardware tokens. Otherwise, the list will contain either a single hardware token (if a match was found) or no hardware tokens. Requires "Grant resource - Read" API permission.

```
GET  /admin/v1/tokens
```

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100` ; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

Parameter	Required?	Description
`type`	Optional*	Specify a type and serial number to look up a single hardware token. One of:

Type	Description
"h6"	HOTP-6 hardware token
"h8"	HOTP-8 hardware token
"yk"	YubiKey AES hardware token
"d1"	Duo-D100 hardware token

		* This option is required if `serial` is present.
`serial`	Optional*	The serial number of the hardware token. * This option is required if `type` is present.

RESPONSE CODES

Response	Meaning
200	Success. Returns a list of tokens.
400	Invalid parameters.

RESPONSE FORMAT

Key	Value
`admins`	A list of administrators associated with this hardware token. See Retrieve Administrators for descriptions of the response fields.
`serial`	The serial number of the hardware token; used to uniquely identify the hardware token when paired with `type`.
`token_id`	The hardware token's unique ID.
`totp_step`	Value is `null` for all supported token types.
`type`	The type of hardware token.
`users`	A list of end users associated with this hardware token. See Retrieve Users for descriptions of the response fields.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": [
    {
      "admins": [
        {
        "admin_id": "DESMP00LIAKRJPD4DZSH",
        "created": 1648143942,
        "email": "jsmith@example.com",
        "last_login": 1343921403,
        "name": "Joe Smith"
        }
    ],
    "serial": "123456",
    "token_id": "DHIZ34ALBA2445ND4AI2",
    "totp_step": null,
    "type": "d1",
    "users": [
      {
      "alias1": "joe.smith",
      "alias2": "jsmith@example.com",
      "alias3": null,
      "alias4": null,
      "aliases": {
        "alias1": "joe.smith",
        "alias2": "jsmith@example.com"
      },
      "created": 1343621411,
      "email": "jsmith@example.com",
```

```
        "enable_auto_prompt": true,
        "firstname": "",
        "is_enrolled": true,
        "last_directory_sync": null,
        "last_login": 1343921403,
        "lastname": "",
        "notes": "",
        "realname": "Joe Smith",
        "status": "active",
        "user_id": "DUJZ2U4L80HT45MQ4EOQ",
        "username": "jsmith"
        }
      ]
    }
  ]
}
```


Create Hardware Token

Create a new hardware token. Requires "Grant resource - Write" API permission.

`POST` `/admin/v1/tokens`

PARAMETERS

Parameter	Required?	Description
`type`	**Required**	The type of hardware token to import. One of: Type: "h6" — HOTP-6 hardware token Type: "h8" — HOTP-8 hardware token Type: "yk" — YubiKey AES hardware token Duo-D100 tokens (type "d1") are imported when purchased from Duo and may not be created via the Admin API.
`serial`	**Required**	The serial number of the token (maximum length 128 characters).
`secret`	Optional	The HOTP secret. This parameter is required for HOTP-6 and HOTP-8 hardware tokens.
`counter`	Optional	Initial value for the HOTP counter. This parameter is only valid for HOTP-6 and HOTP-8 hardware tokens. Default: `0`.
`private_id`	Optional	The 12-character hexadecimal YubiKey private ID. This parameter is required for YubiKey hardware tokens.
`aes_key`	Optional	The 32-character hexadecimal YubiKey AES key. This parameter is required for YubiKey hardware tokens.

RESPONSE CODES

Response	Meaning
200	The hardware token was created successfully. The newly created hardware token is also returned (see Retrieve Hardware Tokens).
400	Invalid or missing parameter(s), or hardware token already exists with the given `type` and `serial`.

RESPONSE FORMAT

Returns the created single token object. Refer to Retrieve Hardware Tokens for an explanation of the object's keys.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "admins": [],
    "serial": "123456",
    "token_id": "DH6G5OP9NU4C5UP00LMN",
    "totp_step": null,
    "type": "h6",
    "users": []
  }
}
```


Retrieve Hardware Token by ID

Return the single hardware token with `token_id`. Requires "Grant resource - Read" or "Grant resource - Write" API permission.

```
GET  /admin/v1/tokens/[token_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No hardware token was found with the given `token_id`.

RESPONSE FORMAT

Returns a single user object. Refer to Retrieve Hardware Tokens for an explanation of the object's keys.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "admins": [
      {
      "admin_id": "DESMP00LIAKRJPD4DZSH",
      "created": 1648143942,
      "email": "jsmith@example.com",
      "last_login": 1343921403,
      "name": "Joe Smith"
      }
    ],
    "serial": "123456",
    "token_id": "DHIZ34ALBA2445ND4AI2",
    "totp_step": null,
    "type": "d1",
    "users": [
      {
      "alias1": "joe.smith",
      "alias2": "jsmith@example.com",
      "alias3": null,
      "alias4": null,
      "aliases": {
```

```
            "alias1": "joe.smith",
            "alias2": "jsmith@example.com"
        },
        "created": 1343621411,
        "email": "jsmith@example.com",
        "enable_auto_prompt": true,
        "firstname": "",
        "is_enrolled": true,
        "last_directory_sync": null,
        "last_login": 1343921403,
        "lastname": "",
        "notes": "",
        "realname": "Joe Smith",
        "status": "active",
        "user_id": "DUJZ2U4L80HT45MQ4EOQ",
        "username": "jsmith"
        }
    ]
  }
}
```


Resync Hardware Token

Resynchronize the hardware token with ID `token_id` by providing three successive codes from the token. Only HOTP and Duo-D100 tokens can be resynchronized. YubiKey tokens operating in their native AES mode do not need resynchronization. Requires "Grant resource - Write" API permission.

`POST` `/admin/v1/tokens/[token_id]/resync`

PARAMETERS

Parameter	Required?	Description
`code1`	**Required**	The first code from the token.
`code2`	**Required**	The second code from the token.
`code3`	**Required**	The third code from the token.

RESPONSE CODES

Response	Meaning
200	The token was resynced successfully.
400	Invalid or missing parameter(s) or cannot resynchronize tokens of this type.
404	No token was found with the given `token_id`.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": ""
}
```


Delete Hardware Token

Delete the hardware token with ID `token_id` from the system. Requires "Grant resource - Write" API permission.

`DELETE` `/admin/v1/tokens/[token_id]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The token was deleted or did not exist.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

U2F Tokens

The U2F Tokens API endpoint `/admin/v1/u2ftokens` is deprecated as of February 2022. This API no longer allows listing all U2F tokens or deletion of U2F tokens. Requests to this endpoint now fail with the following response:

```
{
  "stat": "FAIL",
  "code": 40301,
  "message": "Access forbidden",
}
```

Per-user U2F Token Operations

See Retrieve U2F Tokens by User ID.

WebAuthn Credentials

Per-user WebAuthn Credential Operations

See Retrieve WebAuthn Credentials by User ID.

Retrieve WebAuthn Credentials

Returns a paged list of all registered WebAuthn credentials. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. Requires "Grant resource - Read" API permission to retrieve user credentials, and "Grant administrators - Read" or "Grant administrators - Write" API permission to retrieve administrator credentials.

`GET` `/admin/v1/webauthncredentials`

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned.

		Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	Success. Returns a list of WebAuthn credentials.

RESPONSE FORMAT

Key	Value
`aaguid`	A unique identifier that conveys the authenticator's make and model, or the passkey's provider identity. This value cannot be verified as accurate by Duo.
`admin`	Selected information about the administrator attached to this WebAuthn credential. Returns `null` if attached to an end user. Not returned if the API application does not have "Grant administrators - Read" or "Grant administrators - Write" permission. **Type** / **Description** `admin_id` — The administrator's ID. `created` — The administrator's creation date as a Unix timestamp. `email` — The administrator's email address. `last_login` — An integer indicating the last time this administrator logged in, as a Unix timestamp, or `null` if the administrator has not logged in. `name` — The administrator's name.
`backup_eligible`	If `true`, this credential can be used from multiple devices. If `false`, this credential can only be used on one device.
`backup_status`	If `true`, this credential has been backed up and can be used from multiple devices. If `false`, this credential has not been backed up.
`credential_id`	An identifier randomly generated by the authenticator for this WebAuthn credential.
`credential_name`	Free-form label for this WebAuthn credential.
`date_added`	The Unix timestamp of when this WebAuthn credential was registered in Duo.
`date_last_used`	The Unix timestamp of when this WebAuthn credential was last used to authenticate with Duo. If `null`, this credential has not been used yet.
`label`	A derived nickname for this WebAuthn credential. This value cannot be changed by a user or admin. Present when attached to a user. Example: `Windows Hello` or `iCloud Keychain`.
`passwordless_authorized`	If `true`, this credential can be used for both MFA and Passwordless authentication. If `false`, this credential can only be used for MFA authentication.
`registered_as`	The registration flow that was used to register this WebAuthn credential. One of `platform`, `cross-platform`, or `unknown`.
`transports`	An array of values the browser will use to try and communicate with an authenticator during a Duo authentication attempt.
`user`	Selected information about the end user attached to this WebAuthn credential. See Retrieve Users for descriptions of the response fields. `null` if attached to an

	administrator.
`uv_capable`	If `true`, the authenticator is capable of locally verifying the user's identity. If `false`, the authenticator cannot perform user verification.
`webauthnkey`	The credential's Duo-specific identifier.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": [
    {
        "aaguid": "abcdefgh-123i-4jkl-5mno-6p789012q3rs",
        "admin": null,
        "backup_eligible": true,
        "backup_status": true,
        "credential_id": "A1bCd34efgH5I6jK6lmnoPQrS78",
        "credential_name": "Touch ID",
        "date_added": 1707422111,
        "label": "iCloud Keychain",
        "passwordless_authorized": true,
        "registered_as": "platform",
        "transports": ["hybrid", "internal"],
        "user": {
            "alias1": joesmith,
            "alias2": null,
            "alias3": null,
            "alias4": null,
            "aliases": {},
            "created": 1677007225,
            "email": "",
            "enable_auto_prompt": true,
            "firstname": "",
            "is_enrolled": true,
            "last_directory_sync": null,
            "last_login": 1677007248,
            "lastname": "",
            "notes": "",
            "realname": "",
            "status": "active",
            "user_id": "DUDKPMBF009AHLHNDPDY",
            "username": "jsmith"
        },
        "uv_capable": true,
        "webauthnkey": "WAVTE9JYG4VSQU9LT1B4"
    },
    {
        "aaguid": "abcdefgh-123i-4jkl-5mno-6p789012q3rs",
        "admin": {
            "admin_id": "DEBMWJ05HTBZBU6LNCF3",
            "created": null,
            "email": "ellery.munson@example.com",
            "last_login": 1679420096,
            "name": "Ellery Munson"
        },
        "backup_eligible": false,
        "backup_status": false,
        "credential_id": "A1bCd3_Yx7Xy8Oj_uxJKODn3WFPH7Ml4wqqLIKj5ro",
        "credential_name": "Security key",
        "date_added": 1679413709,
        "date_last_used": 1706714235,
```

```
        "label": "Security key",
        "passwordless_authorized": false,
        "registered_as": "unknown",
        "transports": ["usb"],
        "user": null,
        "uv_capable": false,
        "webauthnkey": "WARRM3HNRDFF3L3AQLXW"
    },
  ]
}
```


Retrieve WebAuthn Credentials by Key

Return the single WebAuthn credential with `webauthnkey`. Requires "Grant resource - Read" API permission to retrieve a user's credential, and "Grant administrators - Read" or "Grant administrators - Write" API permission to retrieve an administrator's credential.

`GET` `/admin/v1/webauthncredentials/[webauthnkey]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No WebAuthn credential was found with the given `webauthnkey` or insufficient API permission to retrieve a credential attached to an administrator.

RESPONSE FORMAT

Returns a single WebAuthn credential object. Refer to Retrieve WebAuthn Credentials for an explanation of the object's keys.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "response": {
        "aaguid": "abcdefgh-123i-4jkl-5mno-6p789012q3rs",
        "admin": null,
        "backup_eligible": true,
        "backup_status": true,
        "credential_id": "A1bCd34efgH5I6jK6lmnoPQrS78",
        "credential_name": "Touch ID",
        "date_added": 1707422111,
        "label": "iCloud Keychain",
        "passwordless_authorized": true,
        "registered_as": "platform",
        "transports": ["hybrid", "internal"],
        "user": {
            "alias1": joesmith,
            "alias2": null,
            "alias3": null,
            "alias4": null,
            "aliases": {},
            "created": 1677007225,
            "email": "",
            "enable_auto_prompt": true,
```

```
            "firstname": "",
            "is_enrolled": true,
            "last_directory_sync": null,
            "last_login": 1677007248,
            "lastname": "",
            "notes": "",
            "realname": "",
            "status": "active",
            "user_id": "DUDKPMBF009AHLHNDPDY",
            "username": "jsmith"
        },
        "uv_capable": true,
        "webauthnkey": "WAVTE9JYG4VSQU9LT1B4"
    }
}
```


Delete WebAuthn Credential

Delete the WebAuthn credential with key `webauthnkey` from the system. Requires "Grant resource - Write" API permission to delete a credential from a user, and "Grant administrators - Write" API permission to delete a credential from an administrator.

`DELETE` `/admin/v1/webauthncredentials/[webauthnkey]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The WebAuthn credential with key `webauthnkey` was deleted.
404	No WebAuthn credential was found with the given `webauthnkey` or insufficient API permission to retrieve a credential attached to an administrator.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": ""
}
```


Desktop Authenticators

Per-user Desktop Authenticators Operations

See Retrieve Desktop Authenticators by User ID.

Retrieve Desktop Authenticators

Returns a paged list of desktop authenticators. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset`. Requires "Grant resource - Read" API permission.

`GET` `/admin/v1/desktop_authenticators`

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	Success.

RESPONSE FORMAT

Key	Value
`daid`	The authenticator's ID.
`dakey`	The authenticator's Duo-specific identifier.
`device_name`	The endpoint's hostname.
`duo_desktop_version`	The version of Duo Desktop installed on the endpoint.
`os_family`	The endpoint's operating system platform.
`user`	Selected information about the end user attached to this Desktop Authenticator. See Retrieve User for descriptions of the response fields.

EXAMPLE RESPONSE


```
{
  "response": [
    {
      "daid": 123,
      "dakey": "DDABCDE1FGHIJ23KL4M",
      "device_name": "RSANCHEZ-M-A12B",
      "duo_desktop_version": "6.12.0.0",
      "os_family": "Mac OS X",
      "user": {
        "alias1": null,
        "alias2": null,
        "alias3": null,
        "alias4": null,
        "aliases": {},
        "created": 1721156264,
        "email": "",
        "enable_auto_prompt": true,
        "firstname": "",
        "is_enrolled": true,
        "last_directory_sync": null,
        "last_login": 1721245422,
        "lastname": "",
        "notes": "",
```

```
      "realname": "",
      "status": "active",
      "user_id": "DABCDEFGHIJKL1MNOPQR",
      "username": "rsanchez"
    }
  },
  {
    "daid": 234,
    "dakey": "DDABCDE1FGHIJ23KL45",
    "device_name": "DESKTOP-ABC1234",
    "duo_desktop_version": "6.12.0",
    "os_family": "Windows",
    "user": {
      "alias1": null,
      "alias2": null,
      "alias3": null,
      "alias4": null,
      "aliases": {},
      "created": 1721156264,
      "email": "",
      "enable_auto_prompt": true,
      "firstname": "",
      "is_enrolled": true,
      "last_directory_sync": null,
      "last_login": 1721245422,
      "lastname": "",
      "notes": "",
      "realname": "",
      "status": "active",
      "user_id": "DABCDEFGHIJKL2MNOPQR",
      "username": "avandalay"
    }
  },
  "stat": "OK"
}
```


Retrieve Desktop Authenticator by Key

Return the single desktop authenticator with `dakey`. Requires "Grant resource - Read" API permission.

```
GET  /admin/v1/desktop_authenticators/[dakey]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No desktop authenticator was found with the given `dakey`.

RESPONSE FORMAT

Returns a single desktop authenticator object. Refer to Retrieve Desktop Authenticators for an explanation of the object's keys.

EXAMPLE RESPONSE

```
{
  "response": {
    "daid": 123,
    "dakey": "DDABCDE1FGHIJ23KL4M",
    "device_name": "RSANCHEZ-M-A12B",
    "duo_desktop_version": "6.12.0.0",
    "os_family": "Mac OS X",
    "user": {
      "alias1": null,
      "alias2": null,
      "alias3": null,
      "alias4": null,
      "aliases": {},
      "created": 1721156264,
      "email": "",
      "enable_auto_prompt": true,
      "firstname": "",
      "is_enrolled": true,
      "last_directory_sync": null,
      "last_login": 1721245422,
      "lastname": "",
      "notes": "",
      "realname": "",
      "status": "active",
      "user_id": "DABCDEFGHIJKL1MNOPQR",
      "username": "rsanchez"
    }
  }
  "stat": "OK"
}
```


Delete Desktop Authenticator

Delete the desktop authenticator with key `dakey` from the system. Requires "Grant resource - Write" API permission to delete a desktop authenticator from a user.

`DELETE` `/admin/v1/desktop_authenticators/[dakey]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The desktop authenticator with key `dakey` was deleted.
404	No desktop authenticator was found with the given `dakey`.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE


```
{
  "response": "",
  "stat": "OK",
}
```

Shared Device Authentication

Retrieve Shared Device Authentication Configurations

Returns Shared Device Authentication configurations. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset`. Requires "Grant resource - Read" API permissions.

```
GET /admin/v1/desktop_authenticators/shared_device_auth
```

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	Success. Returns a list of shared device authentication configurations.

RESPONSE FORMAT

Key	Value
`active`	If `true`, users can authenticate with shared device authentication. Otherwise, `false`.
`groups`	A list of the user groups that are in the shared device authentication configuration. See Retrieve Groups for descriptions of the response fields.
`shared_device_key`	The shared device authentication configuration's Duo-specific identifier.
`name`	The shared device authentication configuration's name.
`trusted_endpoint_integrations`	A list of the management integrations that are in the shared device authentication configuration. **Key** / **Value** `name` — The management integration's name. `trusted_endpoint_integration_id` — The management integration's ID.

EXAMPLE RESPONSE

```
{
  "metadata": {
    "total_objects": 2
  },
  "response": [
```

```
    {
      "active": true,
      "created": "2024-09-03 14:20:52",
      "groups": [
        {
          "desc": "",
          "group_id": "DGABCDE1FGHI2345JKLM",
          "name": "NetAdmins",
          "status": "Active"
        },
        {
          "desc": "",
          "group_id": "DGABCDE1FGHI2345JKL6",
          "name": "Contractors",
          "status": "Active"
        }
      ],
      "shared_device_key": "DDAB1CDEFGHI2JKLMNO",
      "name": "Help Center 1",
      "trusted_endpoint_integrations": [
        {
          "name": "Generic with Duo Desktop",
          "trusted_endpoint_integration_id": "DMAB1CDEF2G34H5IJKLM"
        },
        {
          "name": "Meraki with Duo Desktop",
          "trusted_endpoint_integration_id": "DMAB1CDEF2G34H5IJKL6"
        }
      ]
    },
    {
      "active": false,
      "created": "2024-09-02 14:20:52",
      "groups": [
        {
          "desc": "",
          "group_id": "DGABCDE1FGHI2345JKL6",
          "name": "Help Desk",
          "status": "Active"
        }
      ],
      "shared_device_key": "DDAB1CDEFGHI2JKLMN3",
      "name": "Help Center 2",
      "trusted_endpoint_integrations": [
        {
          "name": "Manual with Duo Desktop",
          "trusted_endpoint_integration_id": "DMAB1CDEF2G34H5IJKL6"
        }
      ]
    }
  ],
  "stat": "OK"
}
```


Retrieve Shared Device Authentication Configuration by Key

Returns a single Shared Device Authentication configuration with `shared_device_key`. Requires "Grant resource - Read" API permissions.

`GET` `/admin/v1/desktop_authenticators/shared_device_auth/[shared_device_key]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No shared device authentication configuration was found with the given `shared_device_key`.

RESPONSE FORMAT

Returns a single shared device authentication configuration. Refer to Retrieve Shared Device Authentication Configurations for an explanation of the object's keys.

EXAMPLE RESPONSE


```
{
  "response": [
    {
      "active": true,
      "created": "2024-09-03 14:20:52",
      "groups": [
        {
          "desc": "",
          "group_id": "DGABCDE1FGHI2345JKLM",
          "name": "NetAdmins",
          "status": "Active"
        },
        {
          "desc": "",
          "group_id": "DGABCDE1FGHI2345JKL6",
          "name": "Contractors",
          "status": "Active"
        }
      ],
      "shared_device_key": "DDAB1CDEFGHI2JKLMNO",
      "name": "Help Center 1",
      "trusted_endpoint_integrations": [
        {
          "name": "Generic with Duo Desktop",
          "trusted_endpoint_integration_id": "DMAB1CDEF2G34H5IJKLM"
        },
        {
          "name": "Meraki with Duo Desktop",
          "trusted_endpoint_integration_id": "DMAB1CDEF2G34H5IJKL6"
        }
      ]
    }
  ],
  "stat": "OK"
}
```


Create Shared Device Authentication Configuration

Create a new shared device authentication configuration with specified management integrations and user groups. Requires "Grant resource - Write" API permission.

`POST` `/admin/v1/desktop_authenticators/shared_device_auth`

PARAMETERS

Parameter	Required?	Description
`group_id_list`	**Required**	A list of one or more group's IDs.
`trusted_endpoint_integration_id_list`	**Required**	A list of one or more management integration's IDs.
`active`	Optional	Specify `1` to allow users to authenticate with shared device authentication. If `0`, shared device authentication is inactive for users. Default: `1`.
`name`	Optional	The shared device authentication configuration's name.

RESPONSE CODES

Response	Meaning
200	Success.
400	The values within `trusted_endpoint_integration_id_list` or `group_id_list` were either incorrect or missing.

RESPONSE FORMAT

Returns the new shared device authentication configuration created. Refer to Retrieve Shared Device Authentication Configurations for an explanation of the object's keys.

EXAMPLE RESPONSE


```
{
  "response": [
    {
      "active": true,
      "created": "2024-09-03 14:20:52",
      "groups": [
        {
          "desc": "",
          "group_id": "DGABCDE1FGHI2345JKLM",
          "name": "NetAdmins",
          "status": "Active"
        },
        {
          "desc": "",
          "group_id": "DGABCDE1FGHI2345JKL6",
          "name": "Contractors",
          "status": "Active"
        }
      ],
      "shared_device_key": "DDAB1CDEFGHI2JKLMNO",
      "name": "Help Center 1",
      "trusted_endpoint_integrations": [
        {
          "name": "Generic with Duo Desktop",
          "trusted_endpoint_integration_id": "DMAB1CDEF2G34H5IJKLM"
        },
        {
          "name": "Meraki with Duo Desktop",
          "trusted_endpoint_integration_id": "DMAB1CDEF2G34H5IJKL6"
        }
      ]
    }
  ],
  "stat": "OK"
}
```

Update Shared Device Authentication Configuration

Update a shared device authentication configuration with a specified `shared_device_key`. Requires "Grant resource - Write" API permission.

Note: This API call uses the PUT request method, which must be represented as JSON and cannot be passed in via URL parameters. Some Duo clients (Python or Java, for example) offer helper functions to handle requests to the Admin API.

```
PUT /admin/v1/desktop_authenticators/shared_device_auth/[shared_device_key]
```

PARAMETERS

Parameter	Required?	Description
`active`	Optional	Specify `1` to allow users to authenticate with shared device authentication. If `0`, shared device authentication is inactive for users.
`group_id_list`	Optional	A list of one or more group's IDs.
`name`	Optional	The shared device authentication configuration's name.
`trusted_endpoint_integration_id_list`	Optional	A list of one or more management integration's IDs.

RESPONSE CODES

Response	Meaning
200	Success.
404	No shared device authentication configuration was found with the given `shared_device_key`.

RESPONSE FORMAT

Returns the updated shared device authentication configuration. Refer to Retrieve Shared Device Authentication Configurations for an explanation of the object's keys.

EXAMPLE RESPONSE


```
{
  "response": [
    {
      "active": true,
      "created": "2024-09-03 14:20:52",
      "groups": [
        {
          "desc": "",
          "group_id": "DGABCDE1FGHI2345JKLM",
          "name": "NetAdmins",
          "status": "Active"
        },
        {
          "desc": "",
          "group_id": "DGABCDE1FGHI2345JKL6",
          "name": "Contractors",
          "status": "Active"
        }
      ],
      "shared_device_key": "DDAB1CDEFGHI2JKLMNO",
      "name": "Help Center 1",
      "trusted_endpoint_integrations": [
        {
          "name": "Generic with Duo Desktop",
          "trusted_endpoint_integration_id": "DMAB1CDEF2G34H5IJKLM"
```

```
        },
        {
          "name": "Meraki with Duo Desktop",
          "trusted_endpoint_integration_id": "DMAB1CDEF2G34H5IJKL6"
        }
      ]
    }
  ],
  "stat": "OK"
}
```


Delete Shared Device Authentication Configuration

Delete a shared device authentication configuration with a specified `shared_device_key`. Requires "Grant resource - Write" API permission.

`DELETE` `/admin/v1/desktop_authenticators/shared_device_auth/[shared_device_key]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The shared device authentication configuration was deleted.
404	No shared device authentication configuration was found with the given `shared_device_key`.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE


```
{
  "response": "",
  "stat": "OK"
}
```


Bypass Codes

Per-user Bypass Code Operations

See Create Bypass Codes for User and Retrieve Bypass Codes by User ID.

Retrieve Bypass Codes

Returns a paged list of information about all bypass codes. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. Output does not include the actual bypass codes. Requires "Grant resource - Read" API permission.

`GET` `/admin/v1/bypass_codes`

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description

`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	Success. Returns metadata information for all bypass codes.

RESPONSE FORMAT

Key	Value
`admin_email`	The email address of the Duo administrator who created the bypass code.
`bypass_code_id`	The bypass code's identifier.
`created`	The bypass code creation date timestamp.
`expiration`	An integer indicating the expiration timestamp of the bypass code, or `null` if the bypass code does not expire on a certain date.
`reuse_count`	An integer indicating the number of times the bypass code may be used before expiring, or `null` if the bypass code has no limit on the number of times it may be used.
`user`	Selected information about the user attached to the bypass code. See Retrieve Users for descriptions of the response fields.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": [
    {
  "admin_email": "janesmith@example.com",
  "bypass_code_id": "DB2A9F0012RL54001FA3",
  "created": 1522260759,
  "expiration": 1522264359,
  "reuse_count": 1,
  "user": {
    "alias1": "joe.smith",
    "alias2": "jsmith@example.com",
    "alias3": null,
    "alias4": null,
    "aliases": {
      "alias1": "joe.smith",
      "alias2": "jsmith@example.com"
    },
    "created": 1384275337,
    "email": "jsmith@example.com",
    "enable_auto_prompt": true,
    "external_id": "1a2345b6-7cd8-9e0f-g1hi-23j45kl6m789",
    "firstname": "",
    "is_enrolled": true,
    "last_directory_sync": 1384275337,
```

```
      "last_login": 1514922986,
      "lastname": "",
      "notes": "",
      "realname": "Joe Smith",
      "status": "active",
      "user_id": "DU3RP9I2WOC59VZX672N",
      "username": "jsmith"
    }
  ]
}
```


Retrieve Bypass Code by ID

Return information about a single bypass code with `bypass_code_id`. Output does not include the actual bypass code. Requires "Grant resource - Read" API permission.

`GET` `/admin/v1/bypass_codes/[bypass_code_id]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No bypass code was found with the given `bypass_code_id`.

RESPONSE FORMAT

Returns a single bypass code object. Refer to Retrieve Bypass Codes for an explanation of the object's keys.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "admin_email": "janesmith@example.com",
    "bypass_code_id": "DB2A9F0012RL54001FA3",
    "created": 1522260759,
    "expiration": 1522264359,
    "reuse_count": 1,
    "user": {
      "alias1": "joe.smith",
      "alias2": "jsmith@example.com",
      "alias3": null,
      "alias4": null,
      "aliases": {
        "alias1": "joe.smith",
        "alias2": "jsmith@example.com"
      },
      "created": 1384275337,
      "email": "jsmith@example.com",
      "enable_auto_prompt": true,
      "external_id": "1a2345b6-7cd8-9e0f-g1hi-23j45kl6m789",
      "firstname": "",
      "is_enrolled": true,
      "last_directory_sync": 1384275337,
      "last_login": 1514922986,
      "lastname": "",
```

```
      "notes": "",
      "realname": "Joe Smith",
      "status": "active",
      "user_id": "DU3RP9I2WOC59VZX672N",
      "username": "jsmith"
    }
  }
}
```


Delete Bypass Code

Delete the bypass code with ID `bypass_code_id` from the system. Requires "Grant resource - Write" API permission.

`DELETE` `/admin/v1/bypass_codes/[bypass_code_id]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The bypass code was deleted.
404	No bypass code was found with the given `bypass_code_id`.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": ""
}
```


Integrations

The Integrations v2 API does not use the same authentication and request signing detailed earlier in this document. Our Python, Go, Java, and Node.JS API clients have been updated with the new authentication and signing requirements and include support for the Integrations v2 API endpoints. We recommend you use the duo_client_python Python API client, the duo_api_golang Go API client, the duo_client_java Java API client, or the duo_api_nodejs Node.JS API client to interact with the Integrations v2 endpoints.

Retrieve Integrations

Returns a paged list of integrations. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. Requires "Grant resource - Read" API permission.

`GET` `/admin/v2/integrations`

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned.

		Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

RESPONSE CODES

Response	Meaning
200	Success.

RESPONSE FORMAT

Key	Value
`adminapi_admins`	Integer value of `1` if the integration has been granted permission to create, modify, and delete Administrators and Administrative Units objects; otherwise `0`. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`adminapi_admins_read`	Integer value of `1` if the integration has been granted permission to read Administrators and Administrative Units objects; otherwise `0`. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`adminapi_allow_to_set_permissions`	Integer value of `1` if the integration has been granted permission to change the permissions for Admin API integrations via the API. If set to `0` then permissions for Admin API applications must be set in the Duo Admin Panel. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`adminapi_info`	Integer value of `1` if the integration has been granted permission to read Account Info; otherwise `0`. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`adminapi_integrations`	Integer value of `1` if the integration has been granted permission to create, modify, and delete Integrations objects; otherwise `0`. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`adminapi_read_log`	Integer value of `1` if the integration has been granted permission to read Logs information; otherwise `0`. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`adminapi_read_resource`	Integer value of `1` if the integration has been granted permission to retrieve objects like users, policies, phones, and hardware tokens; otherwise `0`. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`adminapi_settings`	Integer value of `1` if the integration has been granted permission to read and modify Settings information; otherwise `0`. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`adminapi_write_resource`	Integer value of `1` if the integration has been granted permission to modify and delete objects like users, policies, phones, and hardware tokens;

	otherwise `0`. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`enroll_policy`	Legacy parameter; no effect if specified and returns no value. Use Duo New User policies to configure this setting.
`frameless_auth_prompt_enabled`	Integer value of `1` if the integration has been updated to support Duo Universal Prompt, otherwise `0`. Only appears for a given integration after Duo makes the frameless prompt available for that application, and the value is set to `1` automatically when Duo detects a frameless authentication for the integration.
`greeting`	Voice greeting read before the authentication instructions to users who authenticate with a phone callback.
`groups_allowed`	A list of groups, as group IDs, that are allowed to authenticate with the integration. If empty, all groups are allowed.
`integration_key`	Integration ID.
`ip_whitelist`	Legacy parameter; no effect if specified and always returns an empty list. Use Duo Authorized Network policies to configure this for an application.
`ip_whitelist_enroll_policy`	Legacy parameter; no effect if specified and always returns no value. Use Duo Authorized Network policies to configure this for an application.
`missing_web_referer_policy`	Legacy parameter; no effect if specified and always returns deny.
`name`	The integration's name.
`networks_for_api_access`	A comma-separated list of IP addresses, IP ranges, or CIDRs specifying the networks allowed to access this API integration. Only returned for Accounts API and Admin API integrations.
`notes`	Description of the integration.
`policy_key`	The identifying policy key for the custom policy attached to the integration. Not shown if no policy attached to the integration.
`prompt_v4_enabled`	Integer value of `1` if Duo Universal Prompt is activated for the application, otherwise `0`. Only appears for a given integration when `frameless_auth_prompt_enabled` is `1` (value set automatically when Duo detects a frameless authentication for the integration).
`secret_key`	Secret used when configuring systems to use this integration. The secret key will be hidden, showing only the last four characters.
`self_service_allowed`	Integer value of `1` if users may use self-service from this integration's 2FA prompt to update authentication devices, otherwise `false` (default). Supported on integrations that display the interactive traditional Duo Prompt or the Duo Universal Prompt in a web broswer.
`sso`	Information about your SSO integration. Only returned for SSO integrations. Refer to SSO Parameters for a list of valid values.

`trusted_device_days`	Legacy parameter; no effect if specified and always returns no value. Use Duo Remembered Devices policies to configure this for an application.
`type`	Integration type. One of

Type	Description
"1password"	1Password
"accountsapi"	Accounts API
"adfs"	Microsoft ADFS
"adminapi"	Admin API
"aeries"	Aeries SIS
"agadobe_documentcloud"	DAG - Adobe Document Cloud
"agaha"	DAG - Aha!
"agasana"	DAG - Asana
"agaws"	DAG - Amazon Web Services
"agatlassian-cloud"	DAG - Atlassian Cloud
"agbamboohr"	DAG - BambooHR
"agbarracuda-waf"	DAG - Barracuda WAF
"agbluejeans"	DAG - BlueJeans
"agbomgar"	DAG - Bomgar
"agbonusly"	DAG - Bonusly
"agbox"	DAG - Box
"agbugsnag"	DAG - Bugsnag
"agcanvas"	DAG - Canvas
"agciscoasa"	DAG - Cisco ASA
"agclarizen"	DAG - Clarizen
"agcloudlock"	DAG - CloudLock
"agconfluence"	DAG - Confluence
"agcrashplan"	DAG - CrashPlan
"agcyberark"	DAG - CyberArk Privileged Account Security
"agdatadog"	DAG - Datadog
"agdesk"	DAG - Desk
"agdigicert"	DAG - DigiCert
"agdng"	DAG - Duo Network Gateway
"agdocusign"	DAG - DocuSign
"agdropbox"	DAG - Dropbox
"agduo-adminpanel"	DAG - Duo Admin Panel
"agegnyte"	DAG - Egnyte
"agevernote"	DAG - Evernote
"agexpensify"	DAG - Expensify
"agfacebook"	DAG - Workplace by Facebook
"agfreshdesk"	DAG - Freshdesk

"aggeneric"	DAG - Service Provider
"aggithub-business"	DAG - GitHub.com for Business
"aggithub-enterprise"	DAG - GitHub Enterprise
"aggoogle"	DAG - Google Workspace
"aggotomeeting"	DAG - GoToMeeting
"aggreenhouse"	DAG - Greenhouse
"aghackerone"	DAG - HackerOne
"aghackerrank"	DAG - HackerRank for Work
"agheroku"	DAG - Heroku
"aghipchat"	DAG - HipChat
"agigloo"	DAG - Igloo
"agintacct"	DAG - Intacct
"agjamf-jss"	DAG - Jamf Pro
"agjira"	DAG - JIRA
"agjitbit"	DAG - Jitbit
"aglooker"	DAG - Looker
"agmarketo"	DAG - Marketo
"agmeraki"	DAG - Meraki
"agmonday"	DAG - Monday
"agnamely"	DAG - Namely
"agnetdocuments"	DAG - NetDocuments
"agnewrelic"	DAG - New Relic
"agoffice365"	DAG - Office 365
"agopendns"	DAG - Cisco Umbrella
"agpagerduty"	DAG - PagerDuty
"agpaloalto-aperture"	DAG - Palo Alto Networks Aperture
"agpaloalto"	DAG - Palo Alto Networks
"agremedyforce"	DAG - Remedyforce
"agringcentral"	DAG - RingCentral
"agrobin"	DAG - Robin
"agsalesforce"	DAG - Salesforce
"agsamanage"	DAG - Samanage
"agsaucelabs"	DAG - Sauce labs
"agsharefile"	DAG - ShareFile
"agsignalsciences"	DAG - Signal Sciences
"agslack"	DAG - Slack
"agsmartsheet"	DAG - Smartsheet
"agstatuspageio"	DAG - StatusPage.io
"agsugarcrm"	DAG - SugarCRM
"agsumologic"	DAG - Sumo Logic

"agsyncplicity"	DAG - Syncplicity
"agtableau-online"	DAG - Tableau Online
"agtableau"	DAG - Tableau
"agudemy"	DAG - Udemy
"aguservoice"	DAG - UserVoice
"agwebex"	DAG - Cisco Webex Meetings (with Site Admin)
"agwebex-controlhub"	DAG - Cisco Webex (with Control Hub)
"agworkday"	DAG - Workday
"agzendesk"	DAG - Zendesk
"agzoom"	DAG - Zoom
"akamai-eaa"	Akamai Enterprise Application Access
"array"	Array SSL VPN
"aws-directory-service"	AWS Directory Service
"authapi"	Auth API
"azure-ca"	Microsoft Azure Active Directory
"barracuda"	Barracuda SSL VPN
"bitium"	Bitium
"bitwarden"	Bitwarden
"bomgar"	Bomgar
"caradigm"	Caradigm
"cas"	CAS (Central Authentication Service)
"checkpoint"	Check Point VPN
"cisco"	Cisco ASA SSL VPN
"ciscofirepower"	Cisco Firepower Threat Defense VPN
"ciscoiseadminapi"	Cisco ISE Admin API
"ciscoiseauthapi"	Cisco ISE Auth API
"ciscoiseradius"	Cisco ISE
"ciscoradius"	Cisco RADIUS VPN
"citrixcag"	Citrix Access Gateway
"citrixns"	NetScaler
"clearpass"	Aruba ClearPass
"confluence"	Confluence
"cyberark"	CyberArk Privileged Account Security LDAP/RADIUS
"cyberarkweb"	CyberArk Privileged Account Security
"dag"	Duo Access Gateway Launcher
"device"	Device API
"device-management-portal"	Device Management Portal
"dng"	Duo Network Gateway - Web Application
"dng-rdp"	Duo Network Gateway - RDP Relay

"dng-smb"	Duo Network Gateway - SMB Relay
"dng-ssh"	Duo Network Gateway - SSH Relay
"drawbridgenetworks"	OPAQ 360
"drupal"	Drupal
"epic"	Epic Hyperspace
"f5bigip"	F5 BIG-IP APM
"f5bigipweb"	F5 BIG-IP APM Web
"f5firepass"	F5 FirePass SSL VPN
"fortinet"	Fortinet FortiGate SSL VPN
"greyheller"	Appsian Security Platform
"huntress"	Huntress
"jira"	JIRA
"juniper"	Juniper SSL VPN
"juniperuac"	Juniper UAC
"keeper"	Keeper Security
"labtech"	LabTech Software
"lastpass"	LastPass
"ldapproxy"	LDAP Proxy
"macos"	macOS
"merakiradius"	Meraki RADIUS VPN
"microsoft-eam"	Microsoft Entra ID: External Authentication Methods
"myworkdrive"	MyWorkDrive
"netmotion"	NetMotion Mobility
"oam"	Oracle Access Manager
"okta"	Okta
"onelogin"	OneLogin
"openvpn"	OpenVPN
"openvpnas"	OpenVPN Access Server
"owa"	Microsoft OWA
"paloalto"	Palo Alto SSL VPN
"partner_authapi"	Partner Auth API
"partner_websdk"	Partner WebSDK
"pingfederate"	PingFederate
"portal"	User Self-enrollment Portal (Bulk and Email enrollment)
"radius"	RADIUS
"rdgateway"	Microsoft RD Gateway
"rdp"	Microsoft RDP
"rdweb"	Microsoft RD Web
"resilient"	Resilient Systems

"rest"	Auth API
"rras"	Microsoft RRAS
"sailpoint"	SailPoint API
"sailpointweb"	SailPoint Web
"samlidp"	SAML IdP
"shibboleth"	Shibboleth
"securex-dashboard"	Cisco SecureX Dashboard
"sonicwallsra"	SonicWALL SRA SSL VPN
"sophosutm"	Sophos UTM
"splunk"	Splunk
"sso-generic"	Generic SAML Service Provider
"sso-oauth-client-credentials"	OAuth 2.0 Client Credentials
"sso-oidc-generic"	Generic OIDC Relying Party
"thycotic"	Delinea Secret Server
"tmg"	Microsoft TMG
"uag"	Microsoft UAG
"unix"	Unix Application
"verify"	Verify API
"vmwareview"	VMWare View
"websdk"	Web SDK
"workday"	Workday
"wordpress"	WordPress

`username_normalization_policy`

This controls whether or not usernames should be altered before trying to match them to a user account. Always blank for SSO integrations. One of:

Policy	Description
"None"	The username will be used as is.
"Simple"	Both "DOMAIN\username" and "username@example.com" will be normalized to "username" when logging in.

`web_referers_enabled`

Integer value of `1` if allowed hostnames has been configured; otherwise `0`. Only configurable for integrations that still use the traditional Duo Prompt in an iframe.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": [
    {
      "adminapi_admins": 0,
      "adminapi_admins_read": 0,
      "adminapi_allow_to_set_permissions": 0,
      "adminapi_info": 0,
      "adminapi_integrations": 0,
```

```
    "adminapi_read_log": 0,
    "adminapi_read_resource": 0,
    "adminapi_settings": 0,
    "adminapi_write_resource": 0,
    "enroll_policy": "",
    "frameless_auth_prompt_enabled": 1,
    "greeting": "Welcome to Duo.",
    "groups_allowed": [],
    "integration_key": "DIRWIH0ZZPV4G88B37VQ",
    "ip_whitelist": [],
    "ip_whitelist_enroll_policy": "",
    "missing_web_referer_policy": "deny",
    "name": "Generic SAML Service Provider",
    "notes": "",
    "policy_key": "POHSLA8SP00LABDAD98Y",
    "prompt_v4_enabled": 1,
    "secret_key": "**********************************HazY7",
    "self_service_allowed": false,
    "sso": {
      "idp_metadata": {
        "cert": "-----BEGIN CERTIFICATE-----MIIDDTCCAfW...gAwIBAgIUAdARSAE-----END CERTIFICAT
        "entity_id": "https://sso-abc1def2.sso.duosecurity.com/saml2/idp/RI6WF1LHX9N8GBOEPGZR
        "metadata_url": "https://sso-abc1def2.sso.duosecurity.com/saml2/idp/RI6WF1LHX9N8GBOEP
        "slo_url": "https://sso-abc1def2.sso.duosecurity.com/saml2/idp/RI6WF1LHX9N8GBOEPGZR/s
        "sso_url": "https://sso-abc1def2.sso.duosecurity.com/saml2/idp/RI6WF1LHX9N8GBOEPGZR/s
      },
      "saml_config": {
        "acs_urls": [],
        "assertion_encryption_algorithm": null,
        "attribute_transformations": [],
        "encrypt_assertion": false,
        "entity_id": "",
        "key_transport_encryption_algorithm": null,
        "mapped_attrs": {},
        "nameid_attribute": "",
        "nameid_format": "urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified",
        "relaystate": null,
        "remote_cert": null,
        "role_attrs": {},
        "sign_assertion": true,
        "sign_response": true,
        "signing_algorithm": "http://www.w3.org/2001/04/xmldsig-more#rsa-sha256",
        "slo_url": "",
        "spinitiated_url": null,
        "static_attrs": null
      }
    },
    "trusted_device_days": 0,
    "type": "websdk",
    "username_normalization_policy": "None",
    "web_referers_enabled": 0
  },
  {
    "adminapi_admins": 0,
    "adminapi_admins_read": 0,
    "adminapi_allow_to_set_permissions": 0,
    "adminapi_info": 0,
    "adminapi_integrations": 0,
    "adminapi_read_log": 1,
    "adminapi_read_resource": 0,
    "adminapi_settings": 0,
    "adminapi_write_resource": 0,
```

```
      "enroll_policy": "",
      "greeting": "Welcome to Duo.",
      "groups_allowed": [],
      "integration_key": "DIRWIH0ZZPV4GJ05H7VQ",
      "name": "Admin API",
      "networks_for_api_access": "",
      "notes": "",
      "secret_key": "**********************************HazY7",
      "self_service_allowed": false,
      "type": "adminapi",
      "username_normalization_policy": "None",
    }
  ]
}
```


Retrieve Integrations (Legacy v1)

Returns a paged list of integrations. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. This v1 API endpoint cannot retrieve Duo Single Sign-On applications and returns integrations' secret keys in plain text. Consider migrating to the v2 endpoint. Requires "Grant resource - Read" API permission.

`GET` `/admin/v1/integrations`

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	Success.

RESPONSE FORMAT

Key	Value
`adminapi_admins`	Integer value of `1` if the integration has been granted permission for Administrators methods; otherwise `0`. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`adminapi_admins_read`	Integer value of `1` if the integration has been granted permission to read Administrators and Administrative Units objects; otherwise `0`. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).

`adminapi_allow_to_set_permissions`	Integer value of `1` if the integration has been granted permission to change the permissions for Admin API integrations via the API. If set to `0` then permissions for Admin API applications must be set in the Duo Admin Panel. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`adminapi_info`	Integer value of `1` if the integration has been granted permission for Account Info methods; otherwise `0`. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`adminapi_integrations`	Integer value of `1` if the integration has been granted permission for Integrations methods; otherwise `0`. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`adminapi_read_log`	Integer value of `1` if the integration has been granted permission for Logs methods; otherwise `0`. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`adminapi_read_resource`	Integer value of `1` if the integration has been granted permission to retrieve objects like users, phones, and hardware tokens; otherwise `0`. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`adminapi_settings`	Integer value of `1` if the integration has been granted permission for Settings methods; otherwise `0`. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`adminapi_write_resource`	Integer value of `1` if the integration has been granted permission to modify objects like users, phones, and hardware tokens; otherwise `0`. Returned for all integration types but only applicable to Admin API integrations (`adminapi` type).
`enroll_policy`	Legacy parameter; no effect if specified and returns no value. Use Duo New User policies to configure this setting.
`frameless_auth_prompt_enabled`	Integer value of `1` if the integration has been updated to support Duo Universal Prompt, otherwise `0`. Only appears for a given integration after Duo makes the frameless prompt available for that application, and the value is set to `1` automatically when Duo detects a frameless authentication for the integration.
`greeting`	Voice greeting read before the authentication instructions to users who authenticate with a phone callback.
`groups_allowed`	A list of groups, as group IDs, that are allowed to authenticate with the integration. If empty, all groups are allowed.
`integration_key`	Integration ID.
`ip_whitelist`	Legacy parameter; no effect if specified and always returns an empty list. Use Duo Authorized Network policies to configure this for an application.
`ip_whitelist_enroll_policy`	Legacy parameter; no effect if specified and always returns no value. Use Duo Authorized Network policies to configure this for an application.
`name`	The integration's name.
`networks_for_api_access`	A comma-separated list of IP addresses, IP ranges, or CIDRs specifying the networks allowed to access this API integration. Only returned for Accounts API and Admin API integrations.
`notes`	Description of the integration.
`policy_key`	The identifying policy key for the custom policy attached to the integration. Not shown if no policy attached to the integration.
`prompt_v4_enabled`	Integer value of `1` if Duo Universal Prompt is activated for the application, otherwise `0`. Only appears for a given integration when `frameless_auth_prompt_enabled` is `1` (value set automatically when Duo detects a frameless authentication for the integration).
`secret_key`	Secret used when configuring systems to use this integration.
`self_service_allowed`	Integer value of `1` if users may use self-service from this integration's 2FA prompt to update authentication devices, otherwise `false` (default).

	Supported integrations display the interactive Duo authentication prompt in a web browser. Note: The v1 API response does not include information about Duo Single Sign-On applications.
`type`	Integration type. Refer to Retrieve Integrations for a list of valid values. Note: The v1 API response does not include information about Duo Single Sign-On applications.
`trusted_device_days`	Legacy parameter; no effect if specified and always returns no value. Use Duo Remembered Devices policies to configure this for an application.
`username_normalization_policy`	This controls whether or not usernames should be altered before trying to match them to a user account. One of: **Policy** / **Description** "None" — The username will be used as is. "Simple" — Both "DOMAIN\username" and "username@example.com" will be normalized to "username" when logging in.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": [
    {
      "adminapi_admins": 0,
      "adminapi_admins_read": 0,
      "adminapi_allow_to_set_permissions": 0,
      "adminapi_info": 0,
      "adminapi_integrations": 0,
      "adminapi_read_log": 0,
      "adminapi_read_resource": 0,
      "adminapi_settings": 0,
      "adminapi_write_resource": 0,
      "enroll_policy": "",
      "frameless_auth_prompt_enabled": 1,
      "greeting": "Welcome to Duo.",
      "groups_allowed": [],
      "integration_key": "DIRWIH0ZZPV4G88B37VQ",
      "ip_whitelist": [],
      "ip_whitelist_enroll_policy": "",
      "name": "Web Application",
      "notes": "",
      "policy_key": "POHSLA8SP00LABDAD98Y",
      "prompt_v4_enabled": 1,
      "secret_key": "QO4ZLqQVRIOZYkHfdPDORfcNf8LeXIbCWwHazY7",
      "self_service_allowed": false,
      "trusted_device_days": 0,
      "type": "websdk",
      "username_normalization_policy": "None"
    },
    {
      "adminapi_admins": 0,
      "adminapi_admins_read": 0,
      "adminapi_allow_to_set_permissions": 0,
      "adminapi_info": 0,
      "adminapi_integrations": 0,
      "adminapi_read_log": 1,
      "adminapi_read_resource": 0,
```

```
      "adminapi_settings": 0,
      "adminapi_write_resource": 0,
      "enroll_policy": "",
      "greeting": "Welcome to Duo.",
      "groups_allowed": [],
      "integration_key": "DIRWIH0ZZPV4GJ05H7VQ",
      "name": "Admin API",
      "networks_for_api_access": "",
      "notes": "",
      "secret_key": "QO4ZLqQVRIOZYkHfdPDORj05h8LeXIbCWwHazY7",
      "self_service_allowed": false,
      "type": "adminapi",
      "username_normalization_policy": "None"
    }
  ]
}
```


Create Integration

Create a new integration. The new integration key and secret key are randomly generated and returned in the response. Requires "Grant applications" API permission.

Note: This API call uses POST data, which must be represented as JSON and cannot be passed in via URL parameters. Some Duo clients (Python or Java, for example) offer helper functions to handle requests to the Admin API.

```
POST  /admin/v2/integrations
```

PARAMETERS

Parameter	Required?	Description
`name`	**Required**	The name of the integration to create.
`type`	**Required**	The type of the integration to create. Refer to Retrieve Integrations for a list of valid values. Duo Single Sign-On applications can only be created using the v2 endpoint. Integrations of type "azure-ca" (Microsoft Azure Active Directory) and "microsoft-eam" (Microsoft Entra ID: External Authentication Methods) may not be created via API. If an integration has reached the Duo end of support then new instances of that integration may not be created with the API.
`adminapi_admins`	Optional	Set to `1` to grant to grant an Admin API integration permission to create, modify, and delete all Administrators and Administrative Units objects. Default: `0`.
`adminapi_admins_read`	Optional	Set to `1` to grant an Admin API integration permission to read Administrators and Administrative Units objects. Default: `0`.
`adminapi_allow_to_set_permissions`	Optional	Set to `1` to grant an Admin API integration permission to change the permissions for Admin API integrations via the API. If left at `0` then permissions for Admin API applications must be set in the Duo Admin Panel. Default: `0`.

`adminapi_info`	Optional	Set to `1` to grant an Admin API integration permission for all Account Info methods. Default: `0`.
`adminapi_integrations`	Optional	Set to `1` to grant an Admin API integration permission for all Integrations methods. Default: `0`.
`adminapi_read_log`	Optional	Set to `1` to grant an Admin API integration permission for all Logs methods. Default: `0`.
`adminapi_read_resource`	Optional	Set to `1` to grant an Admin API integration permission to retrieve objects like users, phones, and hardware tokens. Default: `0`.
`adminapi_settings`	Optional	Set to `1` to grant an Admin API integration permission for all Settings methods. Default: `0`.
`adminapi_write_resource`	Optional	Set to `1` to grant an Admin API integration permission to create, modify, and delete objects like users, phones, and hardware tokens. Default: `0`.
`enroll_policy`	Optional	Legacy parameter; no effect if specified and returns no value. Use Duo New User policies to configure this setting. What to do after an unenrolled user passes primary authentication.
`greeting`	Optional	Voice greeting read before the authentication instructions to users who authenticate with a phone callback.
`groups_allowed`	Optional	A comma-separated list of group IDs that are allowed to authenticate with the integration. If empty, all groups are allowed. Object limits: 100 groups per integration.
`ip_whitelist`	Optional	Legacy parameter; no effect if specified and always returns an empty list. Use Duo Authorized Network policies to configure this for an application.
`ip_whitelist_enroll_policy`	Optional	Legacy parameter; no effect if specified and always returns no value. Use Duo Authorized Network policies to configure this for an application. What to do after a new user from a trusted IP completes primary authentication.
`networks_for_api_access`	Optional	A comma-separated list of IP addresses, IP ranges, or CIDRs specifying the networks allowed to access this API integration. Only applicable to Accounts API and Admin API integrations. A given API integration may apply a network restriction to itself via API; use a different API integration to apply the network restriction, or edit the API application in the Duo Admin Panel GUI.
`notes`	Optional	Description of the integration.
`trusted_device_days`	Optional	Legacy parameter; no effect if specified and always returns `0`. Use Duo Remembered Devices policies to configure this for an application.

`self_service_allowed`	Optional	Set to `1` to grant an integration permission to allow users to manage their own devices. This is only supported by integrations which allow for self service configuration.
`sso`	Optional	If creating an SSO integration, refer to SSO parameters.
`username_normalization_policy`	Optional	Policy for whether or not usernames should be altered before trying to match them to a user account. Refer to Retrieve Integrations for a list of valid values.

RESPONSE CODES

Response	Meaning
200	Success. Returns the newly created integration.
400	Invalid or missing parameters, one-to-many object limit reached, integration already exists with the given `name`, or insufficient Admin API permissions.
500	Other internal error.

RESPONSE FORMAT

Returns the created single integration object. Refer to Retrieve Integrations for an explanation of the object's keys.

EXAMPLE RESPONSE

Same as Retrieve Integration by Integration Key.

Create Integration (Legacy v1)

Create a new integration. The new integration key and secret key are randomly generated and returned in the response. This v1 API endpoint cannot create Duo Single Sign-On applications and returns the integration's secret key in plain text. Consider migrating to the v2 endpoint. Requires "Grant applications" API permission.

`POST` `/admin/v1/integrations`

PARAMETERS

Parameter	Required?	Description
`name`	**Required**	The name of the integration to create.
`type`	**Required**	The type of the integration to create. Refer to Retrieve Integrations for a list of valid values. Duo Single Sign-On applications can only be created using the v2 endpoint. Integrations of type "azure-ca" (Microsoft Azure Active Directory) "microsoft-eam" (Microsoft Entra ID: External Authentication Methods) may not be created via API. If an integration has reached the Duo end of support then new instances of that integration may not be created with the API.
`adminapi_admins`	Optional	Set to `1` to grant to grant an Admin API integration permission to create, modify, and delete all Administrators and Administrative Units objects. Default: `0`.
`adminapi_admins_read`	Optional	Set to `1` to grant an Admin API integration permission to read Administrators and Administrative Units objects. Default: `0`.

`adminapi_allow_to_set_permissions`	Optional	Set to `1` to grant an Admin API integration permission to change the permissions for Admin API integrations via the API. If left at `0` then permissions for Admin API applications must be set in the Duo Admin Panel. Default: `0`.
`adminapi_info`	Optional	Set to `1` to grant an Admin API integration permission for all Account Info methods. Default: `0`.
`adminapi_integrations`	Optional	Set to `1` to grant an Admin API integration permission for all Integrations methods. Default: `0`.
`adminapi_read_log`	Optional	Set to `1` to grant an Admin API integration permission for all Logs methods. Default: `0`.
`adminapi_read_resource`	Optional	Set to `1` to grant an Admin API integration permission to retrieve objects like users, phones, and hardware tokens. Default: `0`.
`adminapi_settings`	Optional	Set to `1` to grant an Admin API integration permission for all Settings methods. Default: `0`.
`adminapi_write_resource`	Optional	Set to `1` to grant an Admin API integration permission to create, modify, and delete objects like users, phones, and hardware tokens. Default: `0`.
`enroll_policy`	Optional	Legacy parameter; no effect if specified and returns no value. Use Duo New User policies to configure this setting. What to do after an unenrolled user passes primary authentication.
`greeting`	Optional	Voice greeting read before the authentication instructions to users who authenticate with a phone callback.
`groups_allowed`	Optional	A comma-separated list of group IDs that are allowed to authenticate with the integration. If empty, all groups are allowed. Object limits: 100 groups per integration.
`ip_whitelist`	Optional	Legacy parameter; no effect if specified and always returns an empty list. Use Duo Authorized Network policies to configure this for an application.
`ip_whitelist_enroll_policy`	Optional	Legacy parameter; no effect if specified and always returns no value. Use Duo Authorized Network policies to configure this for an application. What to do after a new user from a trusted IP completes primary authentication.
`networks_for_api_access`	Optional	A comma-separated list of IP addresses, IP ranges, or CIDRs specifying the networks allowed to access this API integration. Only applicable to Accounts API and Admin API integrations. A given API integration may apply a network restriction to itself via API; use a different API integration to apply the network restriction, or edit the API application in the Duo Admin Panel GUI.
`notes`	Optional	Description of the integration.
`trusted_device_days`	Optional	Legacy parameter; no effect if specified and always returns `0`. Use Duo Remembered Devices policies to configure this for an application.

`self_service_allowed`	Optional	Set to `1` to grant an integration permission to allow users to manage their own devices. This is only supported by integrations which allow for self service configuration.
`username_normalization_policy`	Optional	Policy for whether or not usernames should be altered before trying to match them to a user account. Refer to Retrieve Integrations for a list of valid values.

RESPONSE CODES

Response	Meaning
200	Success. Returns the newly created integration.
400	Invalid or missing parameters, one-to-many object limit reached, integration already exists with the given `name`, or insufficient Admin API permissions.
500	Other internal error.

RESPONSE FORMAT

Returns the created single integration object. Refer to Retrieve Integrations for an explanation of the object's keys.

EXAMPLE RESPONSE

Same as Retrieve Integration by Integration Key.

Retrieve Integration by Integration Key

Return the single integration with `integration_key`. Requires "Grant applications" API permission.

```
GET /admin/v2/integrations/[integration_key]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No integration was found with the given `integration_key`.

RESPONSE FORMAT

Returns the single integration object. Refer to Retrieve Integrations for an explanation of the object's keys.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
      "adminapi_admins": 0,
      "adminapi_admins_read": 0,
      "adminapi_allow_to_set_permissions": 0,
      "adminapi_info": 0,
      "adminapi_integrations": 0,
      "adminapi_read_log": 0,
      "adminapi_read_resource": 0,
      "adminapi_settings": 0,
      "adminapi_write_resource": 0,
      "enroll_policy": "",
```

```
    "frameless_auth_prompt_enabled": 1,
    "greeting": "Welcome to Duo.",
    "groups_allowed": [],
    "integration_key": "DIRWIH0ZZPV4G88B37VQ",
    "ip_whitelist": [],
    "ip_whitelist_enroll_policy": "",
    "missing_web_referer_policy": "deny",
    "name": "Generic SAML Service Provider",
    "notes": "",
    "policy_key": "POHSLA8SP00LABDAD98Y",
    "prompt_v4_enabled": 1,
    "secret_key": "**********************************HazY7",
    "self_service_allowed": false,
    "sso": {
        "idp_metadata": {
          "cert": "-----BEGIN CERTIFICATE-----MIIDDTCCAfW...gAwIBAgIUAdARSAE-----END CERTIFICATI
          "entity_id": "https://sso-abc1def2.sso.duosecurity.com/saml2/idp/RI6WF1LHX9N8GBOEPGZR
          "metadata_url": "https://sso-abc1def2.sso.duosecurity.com/saml2/idp/RI6WF1LHX9N8GBOEP
          "slo_url": "https://sso-abc1def2.sso.duosecurity.com/saml2/idp/RI6WF1LHX9N8GBOEPGZR/s
          "sso_url": "https://sso-abc1def2.sso.duosecurity.com/saml2/idp/RI6WF1LHX9N8GBOEPGZR/s
        },
        "saml_config": {
          "acs_urls": [],
          "assertion_encryption_algorithm": null,
          "attribute_transformations": [],
          "encrypt_assertion": false,
          "entity_id": "",
          "key_transport_encryption_algorithm": null,
          "mapped_attrs": {},
          "nameid_attribute": "",
          "nameid_format": "urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified",
          "relaystate": null,
          "remote_cert": null,
          "role_attrs": {},
          "sign_assertion": true,
          "sign_response": true,
          "signing_algorithm": "http://www.w3.org/2001/04/xmldsig-more#rsa-sha256",
          "slo_url": "",
          "spinitiated_url": null,
          "static_attrs": null
        }
    },
    "trusted_device_days": 0,
    "type": "websdk",
    "username_normalization_policy": "None",
    "web_referers_enabled": 0
  }
}
```


Retrieve Integration by Integration Key (Legacy v1)

Return the single integration with `integration_key`. This v1 API endpoint cannot retrieve Duo Single Sign-On applications and returns the integration's secret key in plain text. Consider migrating to the v2 endpoint. Requires "Grant applications" API permission.

`GET` `/admin/v1/integrations/[integration_key]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No integration was found with the given `integration_key`.

RESPONSE FORMAT

Returns the single integration object. Refer to Retrieve Integrations for an explanation of the object's keys.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "adminapi_admins": 0,
    "adminapi_admins_read": 0,
    "adminapi_allow_to_set_permissions": 0,
    "adminapi_info": 0,
    "adminapi_integrations": 0,
    "adminapi_read_log": 0,
    "adminapi_read_resource": 0,
    "adminapi_settings": 0,
    "adminapi_write_resource": 0,
    "enroll_policy": "",
    "frameless_auth_prompt_enabled": 1,
    "greeting": "Welcome to Duo.",
    "groups_allowed": [],
    "integration_key": "DIRWIH0ZZPV4G88B37VQ",
    "ip_whitelist": [],
    "ip_whitelist_enroll_policy": "",
    "name": "Web Application",
    "notes": "",
    "policy_key": "POHSL88SP00LABDAD98Y",
    "prompt_v4_enabled": 1,
    "secret_key": "QO4ZLqQVRIOZYkHfdPDORfcNf8LeXIbCWwHazY7",
    "self_service_allowed": false,
    "trusted_device_days": 0,
    "type": "websdk",
    "username_normalization_policy": "None"
  }
}
```


Modify Integration

Change the name, greeting, and/or notes of the integration with integration key `integration_key`, or reset its secret key. When modifying an Admin API integration permissions can also be added or removed. Requires "Grant applications" API permission.

Note: This API call uses POST data, which must be represented as JSON and cannot be passed in via URL parameters. Some Duo clients (Python or Java, for example) offer helper functions to handle requests to the Admin API.

`POST` `/admin/v2/integrations/[integration_key]`

PARAMETERS

Parameter	Required?	Description

`adminapi_admins`	Optional	Set to `1` to grant an Admin API integration permission to create, modify, or delete Administrators and Administrative Units objects or `0` to remove access to them.
`adminapi_admins_read`	Optional	Set to `1` to grant an Admin API integration permission to read Administrators and Administrative Units objects or `0` to remove access to them.
`adminapi_allow_to_set_permissions`	Optional	Set to `1` to grant an Admin API integration permission to change the permissions for Admin API integrations via the API or `0` to require that permissions for Admin API applications must be set in the Duo Admin Panel.
`adminapi_info`	Optional	Set to `1` to grant an Admin API integration permission for all Account Info methods or `0` to remove access to them.
`adminapi_integrations`	Optional	Set to `1` to grant an Admin API integration permission for all Integrations methods or `0` to remove access to them.
`adminapi_read_log`	Optional	Set to `1` to grant an Admin API integration permission for all Logs methods or `0` to remove access to them.
`adminapi_read_resource`	Optional	Set to `1` to grant an Admin API integration permission to retrieve objects like users, phones, and hardware tokens or `0` to remove access to them.
`adminapi_settings`	Optional	Set to `1` to grant an Admin API integration permission for all Settings methods or `0` to remove access to them.
`adminapi_write_resource`	Optional	Set to `1` to grant an Admin API integration permission to create, modify, and delete objects like users, phones, and hardware tokens or `0` to remove access to them.
`trusted_device_days`	Optional	Legacy parameter; no effect if specified and always returns `0`. Use Duo Remembered Devices policies to configure this for an application. Refer to Retrieve Integrations for a list of supported integrations.
`enroll_policy`	Optional	Legacy parameter; no effect if specified and returns no value. Use Duo New User policies to configure this setting. New policy for what to do after an unenrolled user passes primary authentication. Refer to Retrieve Integrations for a list of valid values.
`greeting`	Optional	New voice greeting. Will be read before the authentication instructions to users who authenticate with a phone callback.
`groups_allowed`	Optional	A comma-separated list of group IDs that are allowed to authenticate with the integration. If set to an empty string, all groups will be allowed. Object limits: 100 groups per integration.
`ip_whitelist`	Optional	Legacy parameter; no effect if specified and always returns an empty list. Use Duo Authorized Network policies to configure this for an application. Refer to Retrieve Integrations for a list of supported integrations.
`ip_whitelist_enroll_policy`	Optional	Legacy parameter; no effect if specified and always returns no value. Use Duo Authorized Network policies to configure this

		for an application. New policy for what to do after a new user from a trusted IP completes primary authentication. Refer to Retrieve Integrations for a list of valid values.
`name`	Optional	New name for the integration.
`networks_for_api_access`	Optional	A comma-separated list of IP addresses, IP ranges, or CIDRs specifying the networks allowed to access this API integration. Only applicable to Accounts API and Admin API integrations. A given API integration may apply a network restriction to itself via API; use a different API integration to apply the network restriction, or edit the API application in the Duo Admin Panel GUI.
`notes`	Optional	New description of the integration.
`policy_key`	Optional	Specify the "Policy Key" value for a custom policy to attach it to the specified integration. Obtain a policy's key by viewing it in the Duo Admin Panel's "Policies" page or from the output of Retrieve Integrations. Leave the value blank to detach the currently attached policy from an integration.
`prompt_v4_enabled`	Optional	Set to `1` to activate Duo Universal Prompt for the application, or to `0` to revert back to traditional prompt. Only appears for a given integration when `frameless_auth_prompt_enabled` is `1` (value set automatically when Duo detects a frameless authentication for the integration).
`reset_secret_key`	Optional	If set to `1`, resets the integration's secret key to a new, randomly generated value. The new secret key is returned in the return value. Attempting to reset the secret key for the same Admin API integration whose integration key and secret key are used to make this call will return an error. Can not be used with SSO applications.
`self_service_allowed`	Optional	Set to `1` to grant an integration permission to allow users to manage their own devices. This is only supported by integrations which allow for self service configuration.
`sso`	Optional	If modifying an SSO integration, refer to SSO parameters.
`username_normalization_policy`	Optional	New policy for whether or not usernames should be altered before trying to match them to a user account. Refer to Retrieve Integrations for a list of valid values.

RESPONSE CODES

Response	Meaning
200	The integration was modified successfully. Also returns the integration object (see Retrieve Integration by Integration Key).
400	Invalid or missing parameters, one-to-many object limit reached, an integration already exists with the given `name`, insufficient Admin API permissions, or attempting to reset the secret key used to sign this API request.
404	No integration was found with the given `integration_key`.
500	Other internal error.

RESPONSE FORMAT

Returns the modified single integration object. Refer to Retrieve Integrations for an explanation of the object's keys.

EXAMPLE RESPONSE

Same as Retrieve Integration by Integration Key.

Modify Integration (Legacy v1)

Change the name, greeting, and/or notes of the integration with integration key `integration_key`, or reset its secret key. When modifying an Admin API integration permissions can also be added or removed. This v1 API endpoint cannot modifyDuo Single Sign-On applications and returns the integration's secret key in plain text. Consider migrating to the v2 endpoint. Requires "Grant applications" API permission.

```
POST  /admin/v1/integrations/[integration_key]
```

PARAMETERS

Parameter	Required?	Description
`name`	Optional	New name for the integration.
`networks_for_api_access`	Optional	A comma-separated list of IP addresses, IP ranges, or CIDRs specifying the networks allowed to access this API integration. Only applicable to Accounts API and Admin API integrations. A given API integration may apply a network restriction to itself via API; use a different API integration to apply the network restriction, or edit the API application in the Duo Admin Panel GUI.
`adminapi_admins`	Optional	Set to `1` to grant an Admin API integration permission to create, modify, or delete Administrators and Administrative Units objects or `0` to remove access to them.
`adminapi_admins_read`	Optional	Set to `1` to grant an Admin API integration permission to read Administrators and Administrative Units objects or `0` to remove access to them.
`adminapi_allow_to_set_permissions`	Optional	Set to `1` to grant an Admin API integration permission to change the permissions for Admin API integrations via the API or `0` to require that permissions for Admin API applications must be set in the Duo Admin Panel.
`adminapi_info`	Optional	Set to `1` to grant an Admin API integration permission for all Account Info methods or `0` to remove access to them.
`adminapi_integrations`	Optional	Set to `1` to grant an Admin API integration permission for all Integrations methods or `0` to remove access to them.
`adminapi_read_log`	Optional	Set to `1` to grant an Admin API integration permission for all Logs methods or `0` to remove access to them.
`adminapi_read_resource`	Optional	Set to `1` to grant an Admin API integration permission to retrieve objects like users, phones, and hardware tokens or `0` to remove access to them.
`adminapi_settings`	Optional	Set to `1` to grant an Admin API integration permission for all Settings methods or `0` to remove access to them.
`adminapi_write_resource`	Optional	Set to `1` to grant an Admin API integration permission to create and modify objects like users, phones, and hardware tokens or `0` to remove access to them.
`trusted_device_days`	Optional	Legacy parameter; no effect if specified and always returns `0`. Use Duo Remembered Devices policies to configure this for an application. Refer to Retrieve Integrations for a list of supported integrations.
`enroll_policy`	Optional	Legacy parameter; no effect if specified and returns no value. Use Duo New User policies to configure this setting.

		New policy for what to do after an unenrolled user passes primary authentication. Refer to Retrieve Integrations for a list of valid values.
`greeting`	Optional	New voice greeting. Will be read before the authentication instructions to users who authenticate with a phone callback.
`groups_allowed`	Optional	A comma-separated list of group IDs that are allowed to authenticate with the integration. If set to an empty string, all groups will be allowed. Object limits: 100 groups per integration.
`ip_whitelist`	Optional	Legacy parameter; no effect if specified and always returns an empty list. Use Duo Authorized Network policies to configure this for an application. Refer to Retrieve Integrations for a list of supported integrations.
`ip_whitelist_enroll_policy`	Optional	Legacy parameter; no effect if specified and always returns no value. Use Duo Authorized Network policies to configure this for an application. New policy for what to do after a new user from a trusted IP completes primary authentication. Refer to Retrieve Integrations for a list of valid values.
`notes`	Optional	New description of the integration.
`policy_key`	Optional	Specify the "Policy Key" value for a custom policy to attach it to the specified integration. Obtain a policy's key by viewing it in the Duo Admin Panel's "Policies" page or from the output of Retrieve Integrations. Leave the value blank to detach the currently attached policy from an integration.
`prompt_v4_enabled`	Optional	Set to `1` to activate Duo Universal Prompt for the application, or to `0` to revert back to traditional prompt. Only appears for a given integration when `frameless_auth_prompt_enabled` is `1` (value set automatically when Duo detects a frameless authentication for the integration).
`reset_secret_key`	Optional	If set to `1`, resets the integration's secret key to a new, randomly generated value. The new secret key is returned in the return value. Attempting to reset the secret key for the same Admin API integration whose integration key and secret key are used to make this call will return an error. Can not be used with SSO applications.
`self_service_allowed`	Optional	Set to `1` to grant an integration permission to allow users to manage their own devices. This is only supported by integrations which allow for self service configuration.
`username_normalization_policy`	Optional	New policy for whether or not usernames should be altered before trying to match them to a user account. Refer to Retrieve Integrations for a list of valid values.

RESPONSE CODES

Response	Meaning
200	The integration was modified successfully. Also returns the integration object (see Retrieve Integration by Integration Key).
400	Invalid or missing parameters, one-to-many object limit reached, an integration already exists with the given `name`, insufficient Admin API permissions, or attempting to reset the secret key used to sign this API request.
404	No integration was found with the given `integration_key`.

500	Other internal error.

RESPONSE FORMAT

Returns the modified single integration object. Refer to Retrieve Integrations for an explanation of the object's keys.

EXAMPLE RESPONSE

Same as Retrieve Integration by Integration Key.

Delete Integration

> **WARNING: Deleting an integration from Duo can block user logins!**
> Be sure to remove Duo authentication from your product's configuration **before** you delete the corresponding integration.
>
> Depending on the application this could mean uninstalling Duo software from your systems, or updating your device or application settings to no longer include Duo in the authentication process.
>
> There is no way to restore an integration deleted in error with Admin API.

Delete the integration with `integration_key` from the system. Attempting to delete the Admin API integration whose secret key is used to sign this request will return an error. Requires "Grant applications" API permission.

```
DELETE  /admin/v2/integrations/[integration_key]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The integration was deleted or did not exist.
400	Attempting to delete the integration whose secret key was used to sign this API request.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Delete Integration (Legacy v1)

> **WARNING: Deleting an integration from Duo can block user logins!**
> Be sure to remove Duo authentication from your product's configuration **before** you delete the corresponding integration.

> Depending on the application this could mean uninstalling Duo software from your systems, or updating your device or application settings to no longer include Duo in the authentication process.
>
> There is no way to restore an integration deleted in error with Admin API.

Delete the integration with `integration_key` from the system. Attempting to delete the Admin API integration whose secret key is used to sign this request will return an error.

This v1 API endpoint cannot delete Duo Single Sign-On applications.

Consider migrating to the v2 endpoint. Requires "Grant applications" API permission.

```
DELETE  /admin/v1/integrations/[integration_key]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The integration was deleted or did not exist.
400	Attempting to delete the integration whose secret key was used to sign this API request.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Retrieve Secret Key

Retrieve the `skey` for the specified integration. Requires "Grant applications" API permission. Does not work for SSO integrations.

This endpoint does not use the same authentication and request signing detailed earlier in this document. Our Python, Go, Java, and Node.JS API clients have been updated with the new authentication and signing requirements and include support for this endpoint operation. We recommend you use the duo_client_python Python API client, the duo_api_golang Go API client, the duo_client_java Java API client, or the duo_api_nodejs Node.JS API client to perform this operation.

```
GET  /admin/v1/integrations/[integration_key]/skey
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The `skey` was successfully returned.
400	No integration was found with the given `integration_key`.

RESPONSE FORMAT

Returns the `skey`.

EXAMPLE RESPONSE

```
{
  "response": {
    "skey": "Zh5eGmUq9zpfQnyUIu5OL9iWoMMv5ZNmk3zLJ4Ep"
  },
  "stat": "OK"
}
```

Retrieve Client Secret for an OAuth Integration

Retrieve the `client_secret` for the specified OAuth integration. Requires "Grant applications" API permission.

```
GET  /admin/v2/integrations/oauth_cc/[integration_key]/client_secret/[client_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The `client_secret` was successfully returned.
400	Invalid parameters or no client was found with the given `client_id`.
404	No integration was found with the given `integration_key`.

RESPONSE FORMAT

Returns the `client_secret`.

EXAMPLE RESPONSE

```
{
  "response": {
    "client_secret": "Zh5eGmUq9zpfQnyUIu5OL9iWoMMv5ZNmk3zLJ4Ep"
  },
  "stat": "OK"
}
```

Reset Client Secret for an OAuth Integration

Reset the `client_secret` for the specified OAuth integration. Requires "Grant applications" API permission.

```
POST  /admin/v2/integrations/oauth_cc/[integration_key]/client_secret/[client_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The `client_secret` was successfully reset.
400	Invalid parameters or no client was found with the given `client_id`.

404	No integration was found with the given `integration_key`.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "response": {},
  "stat": "OK"
}
```

Retrieve Client Secret for an OIDC Integration

Retrieve the `client_secret` for the specified OIDC integration. Requires "Grant applications" API permission.

`GET` `/admin/v2/integrations/oidc/[integration_key]/client_secret`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The `client_secret` was successfully returned.
404	No integration was found with the given `integration_key`.

RESPONSE FORMAT

Returns the `client_secret`.

EXAMPLE RESPONSE

```
{
  "response": {
    "client_secret": "Zh5eGmUq9zpfQnyUIu5OL9iWoMMv5ZNmk3zLJ4Ep"
  },
  "stat": "OK"
}
```

Reset Client Secret for an OIDC Integration

Reset the `client_secret` for the specified OIDC integration. Requires "Grant applications" API permission.

`POST` `/admin/v2/integrations/oidc/[integration_key]/client_secret`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The `client_secret` was successfully reset.
404	No integration was found with the given `integration_key`.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "response": {},
  "stat": "OK"
}
```

SSO Parameters

Parameter	Required?	Description
`idp_metadata`	Automatically generated	Provider information about Duo Single Sign-On.

Key	Value
`authorize_endpoint_url`	The URL that the relying party will redirect the user's browser to during an authentication request. Only returned for OIDC integrations.
`cert`	The certificate used by the service providers to validate the signature on the SAML response sent by Duo Single Sign-On. Only returned for SAML integrations.
`client_id`	The public identifier of the relying party. Only returned for OIDC integrations.
`client_secret`	The client's password. Only returned when an OIDC integration is created.
`discovery_url`	Basic information about OAuth 2.0 or OpenID provider, such as endpoint URLs and supported configuration methods. Only returned for OAuth and OIDC integrations.
`entity_id`	The global, unique name for Duo Single Sign-On. This is sometimes referred to as "Issuer". Only returned for SAML integrations.
`issuer`	Unique entity related to the OAuth 2.0 or OpenID provider that issues a set of claims. Only returned for OAuth and OIDC integrations.
`jwks_endpoint_url`	URL that returns a list of signing keys to validate the signatures of JWTs signed by the OAuth 2.0 or OpenID provider. Only returned for OAuth and OIDC integrations.
`metadata_url`	This URL can be used by service providers to download the XML metadata from Duo Single Sign-On. Only returned for SAML integrations.
`slo_url`	The logout URL for Duo Single Sign-On. This is sometimes referred to as "SLO URL" or "Logout Endpoint". Only returned for SAML integrations.
`sso_url`	The authentication URL for Duo Single Sign-On. This is sometimes referred to as "SSO URL" or "Login URL". This URL can also be used to start IdP-initiated authentications. Only returned for SAML integrations.

`token_endpoint_url`	URL of the authorization server's token endpoint where clients obtain an access token in exchange for a grant or URL used by the relying party to retrieve the access and id tokens after a user has successfully authenticated to the OpenID provider. Only returned for OAuth and OIDC integrations.
`token_introspection_endpoint_url`	URL of the endpoint that validates an access token and retrieves the meta information surrounding the token or URL where relying party can validate an access token. Only returned for OAuth and OIDC integrations.
`userinfo_endpoint_url`	URL where relying party can present access token to retrieve information about the token. Only returned for OIDC integrations.

`oauth_config`	**Required** (if creating an OAuth 2.0 Client Credentials)	Values specific for OAuth 2.0 Client Credentials. See OAuth Parameters.
`oidc_config`	**Required** (if creating a Generic OIDC Relying Party)	Values specific for Generic OIDC Relying Party. See OIDC Parameters.
`saml_config`	**Required** (if creating a SAML integration)	Values specific for Generic SAML Service Provider. See SAML Parameters.

OAuth Parameters

Parameter	Required?	Description
`clients`	**Required**	List of clients. A client is a third-party application that wants to access a resource. Its privileges are limited by scopes.
`scopes`	**Required**	List of scopes. Scopes are a mechanism to authorize a third-party application to perform only specific operations on behalf of the user.

`clients`

Parameter	Required?	Description
`assigned_scopes_ids`	**Required**	List of UUIDs of assigned scopes. If empty, no scope will be assigned.
`client_id`	**Required**	Unique identifier. Needs to be in UUID format.
`name`	**Required**	Client name.
`client_secret`	Returned in response, not a request parameter	Client's password. Only returned when OAuth integration is created.
`description`	Optional	Client description.

`scopes`

Parameter	Required?	Description
`id`	**Required**	Unique identifier that will be used for assigning scope to a client. Needs to be in UUID format.
`name`	**Required**	Scope name.
`name`	Optional	Scope description.

OIDC Parameters

Parameter	Required?	Description
`grant_types`	**Required**	Specify how a client gets an access or refresh token. A client can be configured one grant type.

Parameter	Required?	Description
`authorization_code`	**Required**	Used to exchange an authorization coc token. Parameter / Requir `access_token_lifespan` / Optiona `allow_pkce_only` / Optiona

`refresh_token`	Optional	Used to exchange a refresh token for a when the access token has expired.
		Parameter / Required?
		`refresh_token_chain_lifespan` / Optional / Ex ch tol `2` nc Mi se `1`
		`refresh_token_single_lifespan` / Optional / To for tol `8` nc Mi se `6`

`redirect_uris`	**Required**	List of redirection URIs to which the response will be sent. This URI can exactly r redirection URI values the client pre-registered at the OpenID Provider or a URI subdomain can be used.
`scopes`	**Required**	Scopes are used by a client during authentication to authorize access to a user' scope returns a set of user attributes, which are called claims. Parameter / Required? / Description `openid` / **Required** / Informs the authorization server that the client is ma Connect request. `email` / Optional / Requests access to the `email` and `email_verif`

		`profile`	Optional	Requests access to the user's default profile claims `family_name`, or `given_name`.
`claim_transformations`	Optional	List of transformed claims and rules for their transformations.		

SAML Parameters

Parameter	Required?	Description
`acs_urls`	**Required**	List of URLs where your service provider receives SAML assertions. Each URL is an object with following parameters: `url` — **Required** — ACS URL. `binding` — Optional — The binding that is used to send the request. `index` — Optional — Index number for a given ACS URL. Enter only if you were instructed to do so by the service provider. `isDefault` — Optional — A boolean describing if this URL is a default one.
`entity_id`	**Required**	The service provider identifier.
`nameid_attribute`	**Required**	The authentication source attribute used to identify the user to the service provider.
`nameid_format`	**Required**	Format of NameID when sent to the service provider.
`sign_assertion`	**Required**	A boolean describing if SAML assertion will be signed.
`sign_response`	**Required**	A boolean describing if SAML response will be signed.
`signing_algorithm`	**Required**	Select the encryption strength supported by your service provider. The most common is SHA-256.
`assertion_encryption_algorithm`	Optional	Algorithm used for assertion encryption.
`attribute_transformations`	Optional	List of transformed IdP attributes and rules for their transformations.
`encrypt_assertion`	Optional	A boolean describing if SAML assertion will be encrypted.
`key_transport_encryption_algorithm`	Optional	Algorithm used for key transport encryption.
`mapped_attrs`	Optional	Dictionary of authentication source attributes mapped to the required names.
`relaystate`	Optional	If your service provider requires a specific RelayState parameter, enter it here.
`remote_cert`	Optional	Certificate for signing.
`role_attrs`	Optional	Dictionary of role attributes mapped to multiple roles.
`slo_url`	Optional	URL where your service provider receives SAML logout responses.
`spinitiated_url`	Optional	URL provided by your service provider that will start a SAML authentication.
`static_attrs`	Optional	Dictionary of new attributes mapped to hard coded static attributes.

Policies

The Policies v2 API does not use the same authentication and request signing detailed earlier in this document. Our Python, Go, Java, and Node.JS API clients have been updated with the new authentication and signing requirements and include support for the Policies v2 API endpoints. We recommend you use the duo_client_python Python API client, the duo_api_golang Go API client, the duo_client_java Java API client, or the duo_api_nodejs Node.JS API client to interact with the Policies endpoints.

Summarize Policies

Returns policy names, keys, and applications or groups to which a policy is applied. Requires "Grant resource - Read" API permission.

```
GET  /admin/v2/policies/summary
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success. Returns all policies for the account and where they are applied (unless response is truncated).
400	Invalid or missing parameters.

RESPONSE FORMAT

Key	Value
`policies`	An array of policy blocks, each containing a `policy_name`, `policy_key`, and information on how that policy has been applied.

Key	Value
`policy_applies_to`	An array of all applications, and groups within applications, to which the policy is applied. If the groups block is shown under an application, then the policy only applies to those groups within that application (not the application itself).

Key	Value
`app_name`	Name of the application (same value as the integration's `name`).
`app_integration_key`	The identifier for the application (same value as the integration's `integration_key`).
`apply_type`	Indicates where a given policy is attached: `app` if the policy is assigned to the application for all users of that app, or `group_app` if the policy is assigned as a group policy effective for certain users of that app.
`group_position`	If the application has group policies applied, this number indicates the policy stacking order for those groups. Higher values are closer to the top of the stack.

	`groups` — An array of one or more group `name` and `group_id` values to which the policy is applied
`policy_key`	The identifier for the policy: 20 alphanumeric characters starting with “PO”.
`policy_name`	The name of the policy.

`policy_count`	The total number of policies in this account.
`response_is_truncated`	True if the response has been truncated due to too much data being returned. Otherwise false (default). The response for this call cannot be paged and will be truncated if too much data is returned.
`warnings`	Contains non-fatal failures & warnings.

EXAMPLE RESPONSE


```
{
        "policies": [
            {
                "policy_applies_to": [
                    {
                        "app_integration_key": "DI29C05TCOFOX9QFZEKI",
                        "app_name": "Acme Corp Office 365",
                        "apply_type": "group_app",
                        "group_position": "0",
                        "groups": [
                            {
                                "group_id": "DGC00OELIG7CQD5AV958",
                                "group_name": "Contractors"
                            }
                        ]
                    }
                ],
                "policy_key": "PO0LORUX8W6GXFX27HR0",
                "policy_name": "12 hours Remembered Devices"
            },
            {
                "policy_applies_to": [
                    {
                        "app_integration_key": "DI29C05TCOFOX9QFZEKI",
                        "app_name": "Acme Corp Office 365",
                        "apply_type": "app"
                    }
                ],
                "policy_key": "PO0LBA709Z49E5XCHW06",
                "policy_name": "7 Days Remembered Devices"
            },
            {
                "policy_applies_to": [
                    {
                        "app_integration_key": "DI38Z0OPBEO4G8WE56OV",
```

```
                "app_name": "Acme Cisco VPN",
                "apply_type": "app"
            },
            {
                "app_integration_key": "DION2T45OKE14FSYN4HN",
                "app_name": "Windows Remote Desktop",
                "apply_type": "app"
            }
        ],
        "policy_key": "POAWMUK0XYZQ6BUPBEJF",
        "policy_name": "Deny All Unenrolled Users"
    },
    {
        "policy_applies_to": [],
        "policy_key": "PO9HIT6SE6TWQPVYZD6A",
        "policy_name": "Bypass MFA and Allow Unenrolled"
    },
    {
        "policy_applies_to": [
            {
                "app_integration_key": "DILSVDEYH66ZT8KIXGR9",
                "app_name": "Acme Corp Office 365",
                "apply_type": "group_app",
                "group_position": "1",
                "groups": [
                    {
                        "group_id": "DGIKLVP7LVU968UCDPVJ",
                        "group_name": "Verified Push Users"
                    }
                ]
            }
        ],
        "policy_key": "POWD3Z4WJIE3CF48A33K",
        "policy_name": "Verified Push"
    }
],
"policy_count": 5,
"response_is_truncated": false,
"warnings": [
    "Policy 'Bypass MFA and Allow Unenrolled' is not applied to any groups or application
]
},
"stat": "OK"
}
```


Retrieve Policies

Retrieve a complete set of all policies. Includes all policy section data for each policy. Requires "Grant resource - Read" API permission.

```
GET /admin/v2/policies
```

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned.

		Default: `50`; Max: `100`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid or missing parameters.

RESPONSE FORMAT

Returns a JSON array of policy objects. Refer to Retrieve Policy by ID for an explanation of the policy object's keys and values. See Policy Section Data for descriptions of all policy section data keys and values.

EXAMPLE RESPONSE

```
[
  {
    "policy_name": "Global Policy",
    "policy_key": "POMY5S1FW9345IEM33BK",
    "is_global_policy": true,
    "sections": {
      # Policy Section data appears here
      # See "Policy Section Data" in this document for more information.
    }
  },
  {
    "policy_name": "Contractor Policy",
    "policy_key": "POMY5S1FW93239EM33BK",
    "is_global_policy": false,
    "sections": {
      # Policy Section data appears here
      # See "Policy Section Data" in this document for more information.
    }
  }
]
```

Retrieve Policy by ID

Return the single policy with the specified `policy_key`. Use `global` as a shortcut for the global policy. Includs all policy section data. Requires "Grant resource - Read" API permission.

```
GET  /admin/v2/policies/[policy_key]
```

```
GET  /admin/v2/policies/global
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid or missing parameters.
404	No policy was found with the given `policy_key`.

RESPONSE FORMAT

Returns a single policy object.

Key	Value
`is_global_policy`	True if this policy is the global policy. Otherwise false (default). The value for this key cannot be changed.
`policy_applies_to`	An array of all applications, and groups within applications, to which the policy is applied. If the groups `block` is shown under an application, then the policy only applies to those groups within that application. See Summarize Policies for a complete listing of all keys in this block.
`policy_key`	The identifier for the policy: 20 alphanumeric characters starting with “PO”, or “global” as a shortcut for the global policy. The value for this key cannot be changed.
`policy_name`	Name of the custom policy, or "Global Policy" for the global policy. Policy names do not have to be unique. This key was previously `name`.
`sections`	The list of all enabled policy sections for the policy_key, with associated keys/values for each section. Each section here corresponds to a named section that appears when editing the policy in the Admin Panel. See Policy Section Data for information on all available sections, keys, and values.

Policy Section Name	Policy UI Section Name
`anonymous_networks`	Anonymous Networks
`authentication_methods`	Authentication Methods
`authentication_policy`	Authentication Policy
`authorized_networks`	Authorized Networks
`browsers`	Browsers
`duo_desktop`	Duo Desktop
`duo_mobile_app`	Duo Mobile App
`full_disk_encryption`	Full Disk Encryption
`mobile_device_biometrics`	Mobile Device Biometrics
`new_user`	New User Policy
`operating_systems`	Operating Systems
`plugins`	Plugins

`remembered_devices`	Remembered Devices
`risk_based_factor_selection`	Risk-Based Factor Selection
`screen_lock`	Screen Lock
`tampered_devices`	Tampered Devices
`trusted_endpoints`	Trusted Endpoints
`user_location`	User Location

The number of sections available will vary based on which Duo edition you have.

- **Essentials** edition contains: `authentication_methods`, `authentication_policy`, `authorized_networks`, `duo_desktop`, `new_user`, `remembered_devices`, and `trusted_endpoints` sections.
- **Advantage** edition contains all sections, with additional options for `authorized_networks`, `duo_desktop`, and `remembered_devices`.
- **Premier** edition contains all sections and all options.

SECTION DATA

See Policy Section Data for descriptions of all policy section data keys and values.

Resulting Policy

Returns the resulting policy for a given user/application pair. The resulting policy is constructed from the top-most sections from the entire stack of policies applied for a given `ikey`. Requires "Grant resource - Read" API permission.

```
GET /admin/v2/policies/calculate
```

PARAMETERS

Parameter	Required?	Description
`user_id`	**Required**	The `user_id` of the user to evaluate, as obtained from the Retrieve Users Admin API endpoint or from the Duo Admin Panel.
`integration_key`	**Required**	The `integration_key` of the application to evaluate, as obtained from the Retrieve Integrations Admin API endpoint or from the Duo Admin Panel.

RESPONSE CODES

Response	Meaning
200	Success. The resulting policy information is returned in the response.
400	Invalid or missing parameters.

RESPONSE FORMAT

Returns information about the resulting policy. The `sections` key has the same format as policy section data returned from other Policies endpoints. See Policy Section Data for descriptions of all policy section data keys and values.

Key	Value
`sections`	Policy Section Data from the resulting policy. Each section contains the following information: **Key** / **Value** `source_policy_key` — The policy key value indicating which policy the section is derived from.
`source_policies`	A list of policies that were combined to form the resulting policy. This list is ordered according to the order of precedence used to determine the resulting policy, with the highest precedence policy coming first. Each entry in the list contains the following information: **Key** / **Value** `policy_key` — The policy key value. `policy_name` — The the policy's name. `policy_type` — One of `global`, `application`, or `group`.

EXAMPLE RESPONSE

```
{
    "sections": {
        # Policy section data appears here
    },
    "source_policies": [
        {
            "policy_key": "POLK595BO6OY8OK565I0",
            "policy_name": "Contractor Policy",
            "policy_type": "group"
        },
        {
            "policy_key": "POFLZVHVBOKR23D4N2F0",
            "policy_name": "Web App Policy",
            "policy_type": "application"
        },
        {
            "policy_key": "POCAZZJBRUUR4Q4DQ388",
            "policy_name": "Global Policy",
            "policy_type": "global"
        }
    ]
}
```

Copy Policy

Copies the specified policy to one or more new custom policies. The new policies created by the call will contain the same settings as the copied policy, but will not be applied anywhere. Use Update Policy to apply the new policies to applications or groups within apps. Requires "Grant resource - Write" API permission.

Note: This API call uses POST data, which must be represented as JSON and cannot be passed in via URL parameters. Some Duo clients (Python or Java, for example) offer helper functions to handle requests to the Admin API.

```
POST  /admin/v2/policies/copy
```

PARAMETERS

Parameter	Required?	Description
`policy_key`	**Required**	The identifier for an existing policy to be copied: 20 alphanumeric characters starting with `PO`, or `global` as a shortcut for the global policy.
`new_policy_names_list`	**Optional**	An array of policy names. The policy specified by `policy_key` will be copied once for each name in the list, and the resulting new policy will be given that name.

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid or missing parameters.
404	Policy key specified does not exist.

RESPONSE FORMAT

Returns the new policies.

EXAMPLE RESPONSE

```
{
    "policies": [ {
        "policy_name": "Policy Name",
        "policy_key": "PO239847239H41344KIN",
        "is_global_policy": false,
        "sections": {
              # Section Data that was changed appears here
         }
    },
    {
        "policy_name": "some policy name 2",
        "policy_key": "POMY5S1FW92342EM33BK",
        "is_global_policy": false,
        "sections": { }
     } ],
     "warnings": [ ]
}
```

Create Policy

Creates a new custom policy with the `policy_name` specified in the parameters. Requires "Grant resource - Write" API permission.

Note: This API call uses POST data, which must be represented as JSON and cannot be passed in via URL parameters. Some Duo clients (Python or Java, for example) offer helper functions to handle requests to the Admin API.

```
POST  /admin/v2/policies
```

PARAMETERS

Parameter	Required?	Description
`policy_name`	**Required**	The name of the policy to create. Policy names do not have to be unique.

		The `name` parameter, available during the Policies API public preview, is no longer supported. Please use `policy_name`.
`apply_to_apps`	Optional	Application assignment information for the new policy. **Key** / **Value** `affect_all_apps`: Values are one of: ◦ `inactive` - (default) Use `_list` keys to specify how policy is applied. ◦ `replace-policy` - Apply this policy to all apps, replacing any existing policy. ◦ `apply-policy` - Apply this policy to all applications that don't have a policy applied already. ◦ `unassign-policy` - Remove this policy from all applications to which it is applied. **Warning**: Setting this key to any value other than `inactive` will make changes to all your applications. `apply_list`: An array of applications (specified with `integration_key`) to which to apply this policy. If an application in the list has another policy applied, that policy is kept. `replace_list`: The array of applications (specified with `integration_key`) to which to apply this policy. **Warning**: If the policy is already applied to applications not in the list, it will be removed from those applications. `unassign_list`: An array of applications (specified with `integration_key`) from which to remove this policy. If an application in the list has another policy applied, that policy is kept.
`apply_to_groups_in_apps`	Optional	Information about which groups (within which applications) to which to apply the policy. See Applying Policy to Groups for more information on constructing a list of group policies to affect. **Key** / **Value** `apply_group_policies_list`: A set of groups to apply this policy to in this app as described by `app_integration_key` and `group_id_list`. `group_policy_apply_order`: Values are one of: ◦ `existing` - (default) If the group policy being applied already exists for this application, keep its place in

		the stack. If it doesn't exist, add it to the top of the stack. ◦ `top` - Place this policy on the top of the group policy stack. ◦ `bottom` - Place this policy on the bottom of the group policy stack.
	`replace_group_policies_list`	A set of groups to apply this policy to in this app as described by `app_integration_key` and `group_id_list`. If you specify a blank array (`[]`) as a value, all groups applied to this policy in this application will be replaced with nothing (that is, unassigned).
	`unassign_group_policies_list`	A set of groups to remove this policy from in this app as described by `app_integration_key` and `group_id_list`.
`sections`	Optional	The list of policy sections to be added, with associated keys/values for each section. See Policy Section Data for all sections and their keys/values.

RESPONSE CODES

Response	Meaning
200	Success. The newly-created policy's unique key appears in the response.
400	Invalid or missing parameters.

RESPONSE FORMAT

Returns the new policy object.

EXAMPLE RESPONSE

The `sections` block will be blank on a newly-created policy if no sections were specified in the "Create Policy" call. Use Update Policy to change an existing policy's name, add or remove policy sections, and update keys and values.


```
{
    "policy_applies_to": [
        {
            "app_integration_key": "DI0SUYB00VPUJWGWC2SH",
            "app_name": "Acme Web App",
            "apply_type": "app"
        }
    ],
    "policy_name": "New Policy",
    "policy_key": "POMY5S1FW9345IEM33BK",
    "is_global_policy": false,
```

```
    "sections": {
      # Section Data appears here if it was included in the Create Policy call.
      # See "Policy Section Data" in this document for more information.
    },
    "warnings": [
    ]
}
```

Update Policies

Update policy section data for all policies or a set of specified `policy_key` values. Requires "Grant resource - Write" API permission.

Note: This API call uses the PUT request method, which must be represented as JSON and cannot be passed in via URL parameters. Some Duo clients (Python or Java, for example) offer helper functions to handle requests to the Admin API.

`PUT` `/admin/v2/policies/update`

PARAMETERS

Parameter	Required?	Description
`policies_to_update`	**Required**	The list of policies to update. **Key** / **Value** `edit_all_policies` — Is `true` if the changes should be applied to all policies. Otherwise `false` (default). `edit_list` — An array of policy keys to apply the changes to. Ignored if `edit_all_policies` is true.
`policy_changes`	**Required**	The list of changes to apply to the policies specified in `policies_to_update`. **Key** / **Value** `sections` — The list of policy sections to be updated, with associated keys/values for each section. See Policy Section Data for all sections and their keys/values. `sections_to_delete` — An array of section names to remove from the specified policies. Note that sections cannot be removed from the global policy.

RESPONSE CODES

Response	Meaning
200	One or more policies were modified successfully. The updated policies are also returned.
400	Invalid or missing parameters.

RESPONSE FORMAT

Returns the updated policy objects and the full set of applications and/or groups to which each one is applied.

See Policy Section Data for an explanation of the object's keys/values shown in the `sections` block. Refer to Summarize Policies for a complete listing of all keys in the `policy_applies_to` block.

EXAMPLE RESPONSE

```
{
    "policies": [{
        "policy_name": "Policy 1",
        "policy_key": "PO239847239H41344KIN",
        "is_global_policy": false,
        "sections": {
              # Section Data that was changed appears here
        },
        "policy_applies_to": [{ }]
    },
    {
        "policy_name": "Policy 2",
        "policy_key": "POMY5S1FW92342EM33BK",
        "sections": {},
        "policy_applies_to": [{
            # format for policies that are applied to groups within an app
            "app_name": "Acme Corp",
            "app_integration_key": "DIRWIH0ZZPV8GJ05H7RM",
            "apply_type": "group_app",
            "group_position": "1",
            "groups": [{
                "group_name": "RBA Users",
                "group_id": "DGPDO71QEP8Q03B9C59S"
        },
    {
                "group_name": "API Users",
                "group_id": "DGPDO7F00P8Q03B9C59S"
        } ]
        } ],
        "warnings": [
        # warnings include:
        # "Update: Unrecognized section: <unknown-section> in $POLICY"
        # "Update: Unknown section to delete: <bogus-section> in $POLICY"
        # "Update: Policy (or key) $ID does not exist"
        ]
    } ]
}
```

Update Policy

For the policy with the specified `policy_key`, update policy data and add or change how the policy is applied. Requires "Grant resource - Write" API permission.

What happens if I...

- Make the Update Policy call without one or more of the parameters describing what to modify: `sections`, `sections_to_delete`, `apply_to_apps`, or `apply_to_groups_in_apps`?
  - The API call makes the changes you specify in the input, but all the parameters are optional.
  - If you make the call with no changes in any of the parameters, the call returns the (unchanged) policy.
- Add a policy `section` (such as `authorized_networks`) to a custom policy?
  - The API enables the section (that is, makes it active in the policy) and sets the valid values you specify. Any key/value pairs not specified for the section will be set to their default values.
  - If you add a blank policy section, it will be enabled for the policy with default values.
- Modify values for an **existing** policy `section`?

- The API changes only the values you specify.

Note: This API call uses the PUT request method. The data passed in must be represented as JSON and cannot be passed in via URL parameters. Some Duo clients (Python or Java, for example) offer helper functions to handle requests to the Admin API.

```
PUT /admin/v2/policies/[policy_key]
```

PARAMETERS

<table>
<tr><th>Parameter</th><th>Required?</th><th>Description</th></tr>
<tr><td>apply_to_apps</td><td>Optional</td><td>Application assignment information for the new policy.
<table><tr><th>Key</th><th>Value</th></tr><tr><td>affect_all_apps</td><td>Values are one of:
inactive - (default) Use _list keys to specify how policy is applied.
replace-policy - Apply this policy to all apps, replacing any existing policy.
apply-policy - Apply this policy to all applications that don’t have a policy applied already.
unassign-policy - Remove this policy from all applications to which it is applied.
Warning: Setting this key to any value other than inactive will make changes to all your applications.</td></tr><tr><td>apply_list</td><td>An array of applications (specified with app_integration_key) to which to apply this policy.
If an application in the list has another policy applied, that policy is kept and the apply has no effect.</td></tr><tr><td>replace_list</td><td>The array of applications (specified with app_integration_key) to which to apply this policy.
Warning: If the policy is already applied to applications not in the list, it will be removed from those applications.</td></tr><tr><td>unassign_list</td><td>An array of applications (specified with integration_key) from which to remove this policy.
If an application in the list has another policy applied, that policy is kept and the unassign has no effect.</td></tr></table></td></tr>
<tr><td>apply_to_groups_in_apps</td><td>Optional</td><td>Information about which groups (within which applications) to which to apply the policy. See Applying Policy to Groups for more information on constructing a list of group policies to affect.
<table><tr><th>Key</th><th>Value</th></tr></table></td></tr>
</table>

		`apply_group_policies_list` — A set of groups to apply this policy to in this app as described by `app_integration_key` and `group_id_list`.
		`group_policy_apply_order` — Values are one of: ◦ `existing` - (default) If the group policy being applied already exists for this application, keep its place in the stack. If it doesn't exist, add it to the top of the stack. ◦ `top` - Place this policy on the top of the group policy stack. ◦ `bottom` - Place this policy on the bottom of the group policy stack.
		`replace_group_policies_list` — A set of groups to apply this policy to in this app as described by `app_integration_key` and `group_id_list`. Any groups already specified will be replaced with this set. If you specify a blank array (`[]`) as a value, all groups applied to this policy in this application will be replaced with nothing (that is, unassigned).
		`unassign_all` — Removes all group assignments for that policy. Either `true` or `false`. If `true`, `apply_group_policies_list`, `replace_group_policies_list`, and `unassign_group_policies_list` can not have entries.
		`unassign_group_policies_list` — A set of groups to remove this policy from in this app as described by `app_integration_key` and `group_id_list`.
`policy_name`	Optional	A new name for a custom policy. Policy names do not have to be unique. The `name` parameter, available during the Policies API public preview, is no longer supported. Please use `policy_name`.
`sections`	Optional	The list of policy sections to be added, with associated keys/values for each section. See Policy Section Data for all sections and their keys/values. If `sections` is blank, the policy is unchanged.
`sections_to_delete`	Optional	An array of section names to remove from this policy. Sections cannot be removed from the global policy.

APPLYING POLICY TO GROUPS

Because policy is applied to and unassigned from groups in the context of an application (`integration`), you will need to specify both an `app_integration_key` and an array of groups (`group_id_list`) when applying policy to groups.

Key	Value
`app_integration_key`	The unique identifier for the application (same value as `integration_key`). See the response to Retrieve Integrations.
`group_id_list`	An array of unique group identifiers. See the response to Retrieve Groups to find `group_id` values.

EXAMPLE INPUT

In this example, the "Update Policy" input will remove group policy assignments.

```
{
    "apply_to_groups_in_apps": {
        "unassign_group_policies_list": [ {
            "app_integration_key": "DIRWIH0ZZPV4GJ05H7VQ",
            "group_id_list": ["DGBDKSSH37KSJ373JKSU", "DGJKSLSH393YSJD93HSD3"]
        },
        {
            "app_integration_key": "DIRWIH0ZZPV4GJ05H7HQ",
            "group_id_list": ["DGBDKSSH37KSJ373JKSU", "DGJKSLSH393YSJD93HSD3"]
        } ]
    }
}
```

RESPONSE CODES

Response	Meaning
200	The policy was modified successfully. The policy object is also returned.
400	Invalid or missing parameters.
404	Policy key specified does not exist.

RESPONSE FORMAT

Returns the updated policy object and the full set of associated applications and/or groups.

See Policy Section Data for descriptions of all policy section data keys and values shown in the `sections` block. Refer to Summarize Policies for a complete listing of all keys in the `policy_applies_to` block.

EXAMPLE RESPONSE

```
{
    "is_global_policy": false,
    "policy_key": "POMY5S1FW92342EM33BK",
    "policy_name": "Modified Policy Name",
    "policy_applies_to": [
    {
        "app_integration_key": "DILSVDEYH6Z008KIXGR9",
        "app_name": "Acme Corp",
        "apply_type": "group_app",
        "group_position": "2",
        "groups": [
            {
                "group_id": "DGPDO71QEP8Q03B9C59S",
```

```
            "group_name": "RBA Users"
        }
    ]
},
{
    "app_integration_key": "DI1K3PT8F00DU4Y9IX0I",
    "app_name": "Acme Auth API",
    "apply_type": "group_app",
    "group_position": "0",
    "groups": [
        {
            "group_id": "DGPDO7F00P8Q03B9C59S",
            "group_name": "API Users"
        }
    ]
},
"sections": {
    # Section Data changed appears here.
    # See "Policy Section Data" in this document for more information.
},
"warnings": [
# Contains non-fatal failures and warnings such as misspelled section names.
]
}
```

Delete Policy

Delete the entire custom policy identified by `policy_key` from the system. Requires "Grant resource - write" API permission.

Warning: Policies deleted using this call are immediately and permanently removed from Duo.

To delete policy sections from a custom policy, use an Update Policy call and specify the section(s) in `sections_to_delete`.

`DELETE` `/admin/v2/policies/[policy_key]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The custom policy was deleted.
400	Invalid parameters.
404	The specified custom policy does not exist.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Policy Section Data

Policy sections are used in constructing Create Policy and Update Policy calls. They are also returned as part of the system response to Retrieve Policies, Retrieve Policy by ID, Create Policy, and Update Policy calls.

The API policy `sections` correspond to the named sections visible in the Duo Admin Panel policy editor, as described in the Retrieve Policy by ID "Response Format" section.

We've noted when a specific Duo edition is required to access a section or to enable some keys or values in the section descriptions below.

Anonymous Networks

Section Name: `anonymous_networks`

Corresponds to: Anonymous Networks

This section is available in the Premier and Advantage editions.

Key	Value
`anonymous_access_behavior`	Defines what happens when a user on an anonymous network attempts to access resources. One of `no-action` (default), `require-mfa`, or `deny`.

Authentication Methods

Section Name: `authentication_methods`

Corresponds to: Authentication Methods

Key	Value
`allowed_auth_list`	Comma-separated list of allowed authentication methods. The list defaults to: `duo-push`, `hardware-token`, `webauthn-platform`, `webauthn-roaming`, and `sms`. If Duo Passwordless is turned on for your account, there are three additional authentication methods available: `duo-push-pwl`, `webauthn-platform-pwl`, and `webauthn-roaming-pwl`.
`auto_retry_sms`	Is `true` if a new SMS passcode will be sent up to 3 times when delivery fails. Otherwise `false` (default). Any retries will use additional telephony credits.
`blocked_auth_list`	Comma-separated list of blocked authentication methods. The list defaults to: `desktop`, `duo-passcode`, and `phonecall`.
`require_verified_push`	Is `true` if the user logging in must verify the push by entering the number provided on their authentication device. Otherwise `false`. Default is `true` for the Essentials edition and `false` for Premier and Advantage editions. Applies if `duo-push` is in the `allowed_auth_list`.
`verified_push_digits`	The number of digits a verified push requires the user to enter. An integer between `3` and `6`, inclusive. Defaults to `3`.

Authentication Policy

Section name: `authentication_policy`

Corresponds to: Authentication Policy

Key	Value
`user_auth_behavior`	Defines the behavior when a user authenticates. One of:

	◦ `enforce` (default): Requires 2FA or enrollment when applicable, unless another policy supersedes it. ◦ `bypass`: Skips 2FA and enrollment, unless another policy supersedes it. ◦ `deny`: Denies authentication to all users. Affects all users when enabled.

Authorized Networks

Section Name: `authorized_networks`

Corresponds to: Authorized Networks

<table>
<tr><th>Key</th><th>Value</th></tr>
<tr><td><code>no_2fa_required</code></td><td>Networks the user can access with no 2FA required.
<table>
<tr><th>Key</th><th>Value</th></tr>
<tr><td><code>ip_list</code></td><td>Comma-separated list of public IP addresses for which 2FA is not required. IP address lists can contain individual IPs, IP ranges and IP ranges in CIDR notation. Example: <code>["192.0.2.8", "198.51.100.0-198.51.100.20", "203.0.113.0/24"]</code></td></tr>
<tr><td><code>require_enrollment</code></td><td>Is <code>true</code> (default) if users logging in from these IP addresses must enroll in Duo. Otherwise <code>false</code>. At least one value must be in the <code>ip_list</code> to change this value.</td></tr>
</table></td></tr>
<tr><td><code>mfa_required</code></td><td>Networks the user can access that require MFA. Available in the Premier and Advantage editions.
<table>
<tr><th>Key</th><th>Value</th></tr>
<tr><td><code>ip_list</code></td><td>Comma-separated list of public IP addresses for which MFA is required. IP address lists can contain individual IPs, IP ranges and IP ranges in CIDR notation. Example: <code>["192.0.2.8", "198.51.100.0-198.51.100.20", "203.0.113.0/24"]</code></td></tr>
</table></td></tr>
<tr><td><code>deny_other_access</code></td><td>Is <code>true</code> if users must log in from IP addresses listed in one of the ip_list keys above. Otherwise <code>false</code> (default). At least one IP address must be in either of the <code>ip_list</code> keys above to change this value. Available in the Premier and Advantage editions.</td></tr>
<tr><td><code>block</code></td><td>Networks from which users will be blocked. Available in the Premier and Advantage editions.
<table>
<tr><th>Key</th><th>Value</th></tr>
<tr><td><code>ip_list</code></td><td>Comma-separated list of public IP addresses for which access is blocked. IP address lists can contain individual IPs, IP ranges and IP ranges in CIDR notation. Example: <code>["192.0.2.8", "198.51.100.0-198.51.100.20", "203.0.113.0/24"]</code></td></tr>
</table></td></tr>
</table>

Browsers

Section Name: `browsers`

Corresponds to: Browsers

This section is available in the Premier and Advantage editions.

Key	Value
`allowed_browsers_list`	Comma-separated list of allowed browsers. Any of: `chrome`, `chrome-mobile`, `edge`, `edge-mobile`, `firefox`, `firefox-mobile`, `ie`, `safari`, `safari-mobile`, or `other-browsers`. Default behavior permits all browsers.
`blocked_browsers_list`	Comma-separated list of blocked browsers. Any of: `chrome`, `chrome-mobile`, `edge`, `edge-mobile`, `firefox`, `firefox-mobile`, `ie`, `safari`, `safari-mobile`, or `other-browsers`. Default: none.
`out_of_date_behavior`	Value is one of `warn-only`, `warn-and-block`, or `no-remediation` (default). This affects all browsers in the `allowed_browsers_list`.
`browser_max_ood_days`	The number of days that a browser may be out of date before access to it is blocked (`out_of_date_behavior` must be `warn-and-block` for the browser to be blocked). Value is one of `0`, `14`, `30` (default), `60`, `90`, `180`, or `365`. Other values are invalid.

Duo Desktop

Section Name: `duo_desktop`

Corresponds to: Duo Desktop

This section affects applications protected by Duo Desktop. Duo Desktop was called Duo Device Health (`device_health_app`) prior to November 2023.

Key	Value
`requires_duo_desktop`	Comma-separated list of operating systems that require Duo Desktop (one or more of `linux`, `macos`, or `windows`). Listing an operating system here is the equivalent of setting the “Require Duo Desktop” policy option to "Require the app" for that OS when editing a policy in the Admin Panel.
`prompt_to_install`	Legacy key; no effect if specified.
`bypass_encryption_for_vm`	(Early Access) Comma-separated list of operating systems that will allow the disk encryption, FileVault, or BitLocker check to be bypassed for virtual machines (one or more of `linux`, `macos`, or `windows`). This key is available in the Premier and Advantage editions.
`enforce_device_id_pinning`	Set to `enforce-enabled` to block authentication from any device that presents the same identifiers as another device previously registered by Duo Desktop. Default: `no-enforcement`. This setting requires `enforce_signed_payload` be set to `enforce-enabled` in the same policy.
`enforce_encryption`	Comma-separated list of operating systems that will require the hard drive to be encrypted (one or more of `linux`, `macos`, or `windows`). This key is available in the Premier and Advantage editions.
`enforce_firewall`	Comma-separated list of operating systems that will require a firewall to be active (one or more of `linux`, `macos`, or `windows`). This key is available in the Premier and Advantage editions.

`enforce_signed_payload`	Set to `enforce-enabled` to require device registration with Duo Desktop during initial authentication. The access device must support TPM 2.0 on Windows or Secure Enclave on macOS. Default: `no-enforcement`.
`enforce_system_password`	Comma-separated list of operating systems that will require a system password to be set (one or more of `linux`, `macos`, or `windows`). This key is available in the Premier and Advantage editions.
`linux_endpoint_security_list`	Comma-separated list of Duo-supported endpoint security agents that are allowed. For agents in this list, the application will block access unless one of those agents is running. A complete list of Linux security agents is available in a drop-down when editing the policy in the Admin Panel. This key is available in the Premier edition.
`linux_remediation_note`	A text note (max 700 characters) with remediation instructions when an end user is blocked. This key is available in the Premier edition.
`macos_endpoint_security_list`	Comma-separated list of Duo-supported endpoint security agents that are allowed. For agents in this list, the application will block access unless one of those agents is running. A complete list of macOS security agents is available in a drop-down when editing the policy in the Admin Panel. This key is available in the Premier edition.
`macos_remediation_note`	A text note (max 700 characters) with remediation instructions when an end user is blocked. This key is available in the Premier edition.
`windows_endpoint_security_list`	Comma-separated list of Duo-supported endpoint security agents that are allowed. For agents in this list, the application will block access unless one of those agents is running. A complete list of Windows security agents is available in a drop-down when editing the policy in the Admin Panel. This key is available in the Premier edition.
`windows_remediation_note`	A text note (max 700 characters) with remediation instructions when an end user is blocked. This key is available in the Premier edition.

Duo Mobile App

Section Name: `duo_mobile_app`

Corresponds to: Duo Mobile App

This section is available in the Premier and Advantage editions.

Key	Value
`block_policy`	Indicates when the user will be blocked from authenticating. Value is one of: ◦ `no-remediation`: No version checking; equivalent to “Never” when editing the policy in the UI. ◦ `less-than-latest`: Duo Mobile app is not at the most recently-released version. ◦ `less-than-version` (default): Duo Mobile app is older than the version specified in `block_version`.
`block_remediation_days`	The number of days before the user will be blocked. Value is one of `0` (default), `14`, `30`, `60`, `90`, `180`, or `365`. Applicable only if `block_policy` is `less-than-latest`.

`block_version`	The specific Duo Mobile app version (from the list in the edit policy UI) that is being blocked. The default is `3.8.0`. Applicable only if `block_policy` is `less-than-version`.
`require_updates`	(Deprecated) Is `true` (default) if the Duo Mobile app must have up-to-date security patches. Otherwise `false`.
`warn_policy`	Indicates when the user should be warned that their Duo Mobile app is out of date. Value is one of: ◦ `no-remediation` (default): No version checking; equivalent to “Never” when editing the policy in the UI. ◦ `less-than-latest`: Duo Mobile app is not at the most recently-released version. ◦ `less-than-version`: Duo Mobile app is older than the version specified in `warn_version`.
`warn_remediation_days`	The number of days that the user will be warned. Value is one of `0`, `14`, `30`, `60`, `90`, `180`, or `365`. Applicable only if `warn_policy` is `less-than-latest`.
`warn_version`	The specific Duo Mobile version (from the list in the edit policy UI) subject to the out-of-date warning. Applicable only if `warn_policy` is `less-than-version`.

Full Disk Encryption

Section Name: `full_disk_encryption`

Corresponds to: Full Disk Encryption

This section is available in the Premier and Advantage editions.

Key	Value
`require_encryption`	Is `true` if the device used for authentication requires full-disk encryption. Otherwise `false` (default).

Mobile Device Biometrics

Section Name: `mobile_device_biometrics`

Corresponds to: Mobile Device Biometrics

This section is available in the Premier and Advantage editions.

Key	Value
`block_biometric_pin_fallback`	Is `true` if the mobile device used to authenticate is not allowed to fall back to the device's passcode when approving Duo Push login requests and biometric verification fails. Otherwise `false` (default).
`require_biometrics`	Is `true` if the mobile device used to authenticate requires Apple Touch ID, Face ID, or Android Fingerprint as additional verification when approving Duo Push login requests. Otherwise `false` (default).

New User Policy

Section Name: `new_user`

Corresponds to: New User Policy

Key	Value

`new_user_behavior`	Controls what happens after an unenrolled user passes primary authentication. One of: ◦ `enroll` (default): Require the user to enroll whenever possible. ◦ `no-mfa`: MFA is not required for unknown users; unenrolled users must enroll. ◦ `deny`: Denies authentication to unenrolled users.

Operating Systems

Section Name: `operating_systems`

Corresponds to: Operating Systems

This section is available in the Premier and Advantage editions.

The `operating_systems` section affects devices used to access applications protected by Duo's browser-based authentication prompt, and mobile devices using Duo Mobile as a second factor.

The operating system names that can be in each of the lists below are: `android`, `blackberry`, `chromeos`, `ios`, `linux`, `macos`, `windows`, `windowsphone`, and `unknownos`.

Note: No operating system should be listed in more than one of `allow_unrestricted_os_list`, `block_os_list`, and `os_restrictions`. In the case of conflicts, the API call will fail.

Key	Value
`allow_unrestricted_os_list`	Comma-separated list of operating systems that are allowed with no constraints or warnings. `allow_unrestricted_os_list` and `block_os_list` must not contain the same values.
`block_os_list`	Comma-separated list of operating systems that are not allowed. `allow_unrestricted_os_list` and `block_os_list` must not contain the same values. Blocked Android or iOS versions will not be able to authenticate using Duo Push or Duo Mobile passcodes.
`os_restrictions`	Can contain up to 4 blocks of keys – one for each operating system that can be further restricted: `android`, `ios`, `macos`, and `windows`.

Key	Value
`warn_policy`	Indicates when the user should be warned that their OS is out of date. Value is one of: ◦ `no-remediation` (default): No version checking; equivalent to “Never” when editing the policy in the UI. ◦ `end-of-life`: OS vendor no longer releases security updates. ◦ `not-up-to-date`: OS is not at the most recent patch release version. ◦ `less-than-latest`: OS is not at the most recently-released version. ◦ `less-than-version`: OS is older than the version specified in `warn_version`.
`warn_version`	The specific OS version (from the list in the edit policy UI) subject to the out-of-date warning. Applicable only if `warn_policy` is `less-than-version`.

`warn_remediation_days`	Number of days that the user will be warned. Value is one of `0`, `14`, `30` (default), `60`, `90`, `180`, or `365`.
`block_policy`	Indicates when the user will be blocked from access. Values are the same as `warn_policy`.
`block_version`	The specific OS version (from the list in the edit policy UI) that is being blocked. Values are the same as `warn_version`. Applicable only if `block_policy` is `less-than-version`.
`block_remediation_days`	Number of days before the user will be blocked. Values are the same as `warn_remediation_days`.

Example section: Warn for two weeks if iOS is not up to date; block after 60 days total.

```
"operating_systems": {
"allow_unrestricted_os_list": [
    "android",
    "chromeos",
    "linux",
    "macos",
    "unknownos",
    "windows"
  ],
  "block_os_list": [
    "blackberry",
    "windowsphone"
  ],
  "os_restrictions": {
      "ios": {
          "block_policy": "not-up-to-date",
          "block_remediation_days": 60,
          "block_version": "",
          "warn_policy": "not-up-to-date",
          "warn_remediation_days": 14,
          "warn_version": ""
      }
  }
}
```

Plugins

Section Name: `plugins`

Corresponds to: Plugins

This section is available in the Premier and Advantage editions.

Key	Value
`flash`	Specify how Flash plugins are treated. Value is one of `allow-all` or `block-all` (default).
`java`	Specify how Java plugins are treated. Value is one of: ◦ `allow-all`: No restrictions. ◦ `warn-only` (default): Warn if plugin is out of date. ◦ `warn-and-block`: warn if out of date; block after `java_max_ood_days`. ◦ `block-all`: Blocked; no access.

`java_max_ood_days`	The number of days that Java plugins may be out of date before access is blocked (`java` must be `warn-and-block` for the plugin to be blocked). Value is one of `0` , `14` , `30` (default), `60` , `90` , `180` , or `365` . Other values are invalid.

Remembered Devices

Section Name: `remembered_devices`

Corresponds to: Remembered Devices

This section affects how and when users can skip subsequent 2FA and Duo Passwordless (if enabled) login requests.

Note: Passwordless users will be able to remember devices for no longer than three days, regardless of the selected length of time entered via `max_time_units` and `max_time_value` .

Key	Value
`browser_apps`	

Key	Value
`enabled`	Is `true` (default) if devices are remembered for browser-based apps. Otherwise `false` .
`remember_method`	One of `user-based` or `risk-based` (default). `risk-based` only available in the Premier and Advantage editions.
`user_based`	Allow users to remember their devices for browser-based applications.

Key	Value
`confirm_per_app`	Is `true` if the user must confirm for each browser-based application separately. Otherwise `false` (default).
`max_time_units`	One of `days` or `hours` (default).
`max_time_value`	If `max_time_units` is set to: ◦ `days` : an integer `1` to `365` , inclusive. ◦ `hours` : an integer `1` to `8760` , inclusive. Defaults to `12` .

`risk_based`	This set of keys is available in the Premier and Advantage editions. Remember user devices using risk-based authentication.

Key	Value
`max_time_units`	One of `days` (default) or `hours` .
`max_time_value`	If `max_time_units` is set to: ◦ `days` : an integer `1` to `365` , inclusive. ◦ `hours` : an integer `1` to `8760` , inclusive. Defaults to `30` .

`windows_logon`

Key	Value
`enabled`	Is `true` if devices are remembered for Windows Logon. Otherwise `false` (default). 2FA will be enforced after users sign out, reboot, or change networks.
`max_time_units`	One of `days` (default) or `hours`.
`max_time_value`	If `max_time_units` is set to: ◦ `days`: an integer `1` to `365`, inclusive. ◦ `hours`: an integer `1` to `8760`, inclusive. Defaults to `30`.

Risk-Based Factor Selection

Section Name: `risk_based_factor_selection`

Corresponds to: Risk-Based Factor Selection

This section is available in the Premier and Advantage editions.

To enable these policy settings, applications must have the Universal Prompt activated or use the named "Duo Auth API" application. Learn more about requirements for these policy settings.

Key	Value
`limit_to_risk_based_auth_methods`	Is `true` (default) if the user is limited to risk-based authentication methods when Duo detects a higher-risk authentication. Otherwise `false`.
`risk_based_verified_push_digits`	The number of digits a verified push requires the user logging in to enter. An integer between `3` and `6`, inclusive. Defaults to `6`.

Screen Lock

Section Name: `screen_lock`

Corresponds to: Screen Lock

This section is available in the Premier and Advantage editions.

Key	Value
`require_screen_lock`	Is `true` (default) if the device must have a screen lock to be allowed for authentication. Otherwise `false`. Applies to iOS (8 and up) and Android.

Tampered Devices

Section Name: `tampered_devices`

Corresponds to: Tampered Devices

This section is available in the Premier and Advantage editions.

Key	Value

`block_tampered_devices`	Is `true` (default) if iOS or Android devices that are rooted or otherwise tampered with are not allowed for authentication. Otherwise `false`.

Trusted Endpoints

Section Name: `trusted_endpoints`

Corresponds to: Trusted Endpoints

A trusted endpoint is an endpoint that exists in a management system such as your EAM or MDM. It can be matched to your management system using Duo certificates or information provided by the Duo Mobile app.

Key	Value
`trusted_endpoint_checking`	Value is one of `allow-all` (default), `require-trusted`, or `not-configured`. Value will be `not-configured` if Trusted Endpoints management systems have not been configured. Trusted endpoints will still be checked for reporting purposes if `allow-all` is set, but untrusted endpoints will be allowed.
`cisco_secure_endpoint_can_block`	Is `true` if Cisco Secure Endpoint is allowed to block compromised endpoints. Otherwise `false` (default).
`trusted_endpoint_checking_mobile`	Allowed values and default are the same as `trusted_endpoint_checking`. Duo recommends not setting this value separately from `trusted_endpoint_checking`. Since the user-agent string is self-reported by the browser, it's possible to manipulate it from the client side to change the value reported to Duo, with the potential effect of bypassing a policy intended to block access.

User Location

Section Name: `user_location`

Corresponds to: User Location

This section is available in the Premier and Advantage editions.

Obtain Alpha-2 country codes from https://www.iso.org/obp/. Choose **Country Codes** and then **Search** to see the list of country codes.

Key	Value
`require_mfa_countries_list`	List of one or more country codes. If the user's location matches one of the codes, that user is required to use MFA to authenticate.
`deny_access_countries_list`	List of one or more country codes. If the user's location matches one of the codes, that user is denied access.
`allow_access_no_2fa_countries_list`	List of one or more country codes. If the user's location matches one of the codes, that user is allowed access without 2FA.
`ignore_location_countries_list`	List of one or more country codes. If the user's location matches one of the codes, no action is taken.
`default_action`	Indicates behavior for country codes that aren't in any list. Values are one of: ◦ `deny-access`: User is denied access. ◦ `ignore-location` (default): No action is taken.

	◦ `require-mfa`: User must use MFA to authenticate. ◦ `allow-access-no-2fa`: User is allowed access without 2FA.

Endpoints

Retrieve Endpoints

Returns a paged list of endpoints. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. When no `limit` is specified the response is limited to 100 results. Requires "Grant resource - Read" API permission.

Information for a given endpoint is purged after 30 days of inactivity.

Endpoint information retrievable by Duo Premier and Duo Advantage customers. In addition, some response information is available only with Duo Premier.

`GET` `/admin/v1/endpoints`

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	Success.

RESPONSE FORMAT

Key	Value

`browsers`	Collected information about all detected browsers on an individual endpoint. **Key** / **Value** `browser_family` — The web browser family. `browser_version` — The web browser version. `flash_version` — The Flash plugin version. If not present then "uninstalled". `java_version` — The Java plugin version. If not present then "uninstalled". `last_used` — The date and time that the endpoint's browser was last used for access, shown as a Unix timestamp.
`computer_sid`	The machine security identifier of a Windows endpoint.
`cpu_id`	The CPU ID of a Windows endpoint.
`device_id`	Custom device identifier of a Meraki-managed iOS endpoint. Returned for Duo Premier customers only.
`device_identifier`	The unique device attribute value that identifies the endpoint. Returned for Duo Premier customers only. This property will be deprecated in a future release.
`device_identifier_type`	The device attribute used to identify a unique endpoint. One of "hardware_uuid", "fqdn", "hardware_serial", "device_udid", or none. This property will be deprecated in a future release.
`device_name`	The endpoint's hostname.
`device_udid`	The unique device identifier for iOS endpoints managed by Workspace ONE, Ivanti Endpoint Manager Mobile, Ivanti Neurons for MDM, or Sophos Mobile via certificates. Returned for Duo Premier customers only.
`device_username`	The unique attribute value that identifies the endpoint's associated user in the management system. Returned for Duo Premier customers only.
`device_username_type`	The management system attribute used to identify the user associated with the unique endpoint. One of "os_username", "upn", "username", "email", or none. Returned for Duo Premier customers only.
`disk_encryption_status`	The hard drive encryption status of the endpoint as detected by Duo Desktop. One of "On", "Off", or "Unknown".
`domain_sid`	The Active Directory domain security identifier for a domain-joined Windows endpoint. Empty if the Windows endpoint is not joined to a domain.
`email`	The email address, if present, of the user associated with an endpoint.
`epkey`	The endpoint's unique identifier. Most reliable when reported by Duo Desktop installed on the endpoint.
`firewall_status`	Status of the endpoint's local firewall as detected by Duo Desktop. One of "On", "Off", or "Unknown".
`hardware_uuid`	The universally unique identifier for a Mac endpoint.
`health_app_client_version`	The version of Duo Desktop installed on the endpoint.
`health_data_last_collected`	The last time Duo Desktop performed a device health check, as a Unix timestamp.
`last_updated`	The last time the endpoint accessed Duo, as a Unix timestamp.
`machine_guid`	The globally unique identifier for a Windows or macOS endpoint.
`model`	The device model of a 2FA endpoint.
`os_build`	The endpoint's operating system build number.
`os_family`	The endpoint's operating system platform.
`os_version`	The endpoint's operating system version.
`password_status`	Whether the local admin password is set on the endpoint as detected by Duo Desktop. One of "Set", "Unset", or "Unknown".

`security_agents`	Information about security agents present on the endpoint as detected by Duo Desktop. Returned for Duo Premier customers only. **Key** / **Value** `security_agent` — The name of the security agent. `version` — The security agent version.
`trusted_endpoint`	Whether the endpoint is a Duo managed endpoint. One of "yes", "no", or "unknown".
`type`	The endpoint's device class.
`username`	The Duo username of the user associated with an endpoint.

EXAMPLE RESPONSE

Example response for a Duo Premier plan customer


```
{
  "stat": "OK",
  "response": [{
        "browsers": [{
            "browser_family": "Chrome",
            "browser_version": "91.0.4472.77",
            "flash_version": "uninstalled",
            "java_version": "uninstalled",
            "last_used": 1624451420
        },
        {
            "browser_family": "Safari",
            "browser_version": "14.1",
            "flash_version": "uninstalled",
            "java_version": "uninstalled",
            "last_used": 1624457297
        }],
        "computer_sid": "",
        "cpu_id": "",
        "device_id": "",
        "device_identifier": "3FA47335-1976-3BED-8286-D3F1ABCDEA13",
        "device_identifier_type": "hardware_uuid",
        "device_name": "ejmac",
        "device_udid": "",
        "device_username": "mba22915\u00e2\u0080\u0099s MacBook Air/mba22915",
        "device_username_type": "os_username",
        "disk_encryption_status": "On",
        "domain_sid": "",
        "email": "ejennings@example.com",
        "epkey": "EP18JX1A10AB102M2T2X",
        "firewall_status": "On",
        "hardware_uuid": "3FA47335-1976-3BED-8286-D3F1ABCDEA13",
        "health_app_client_version": "2.13.1.0",
        "health_data_last_collected": 1624451421,
        "last_updated": 1624451420,
        "machine_guid": "",
        "model": "",
        "os_build": "19H1030",
        "os_family": "Mac OS X",
        "os_version": "10.11.7",
        "password_status": "Set",
        "security_agents": [{
            "security_agent": "Cisco AMP for Endpoints",
```

```
            "version": "10.1.2.3",
        }],
        "trusted_endpoint": "yes",
        "type": "",
        "username": "ejennings"
    },
    {
        "browsers": [{
        {
            "browser_family": "Mobile Safari",
            "browser_version": "9.0",
            "flash_version": "uninstalled",
            "java_version": "uninstalled"
        }],
        "computer_sid": "",
        "cpu_id": "",
        "device_id": "",
        "device_identifier": "",
        "device_identifier_type": "",
        "device_name": "",
        "device_udid": "",
        "device_username": "",
        "device_username_type": "",
        "disk_encryption_status": "Unknown",
        "domain_sid": "",
        "email": "mhanson@example.com",
        "epkey": "EP65MWZWXA10AB1027TQ",
        "firewall_status": "Unknown",
        "hardware_uuid": "",
        "health_app_client_version": "",
        "health_data_last_collected": "",
        "last_updated": 1622036309,
        "machine_guid": "",
        "model": "iPhone",
        "os_build": "",
        "os_family": "iOS",
        "os_version": "14.5.1",
        "password_status": "Unknown",
        "security_agents": [],
        "trusted_endpoint": "unknown",
        "type": "",
        "username": "mhanson"
    }],
}
```


Retrieve Endpoint by ID

Return information for an individual endpoint with `epkey`. Requires "Grant resource - Read" API permission.

Some response information available for Duo Premier customers only.

`GET` `/admin/v1/endpoints/[epkey]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning

200	Success.
404	No endpoint was found with the given `epkey`.

RESPONSE FORMAT

Same as Retrieve Endpoints.

EXAMPLE RESPONSE

Example response for a Duo Advantage plan customer


```
{
  "stat": "OK",
  "response": {
        "browsers": [{
            "browser_family": "Edge Chromium",
            "browser_version": "91.0.864.54",
            "flash_version": "uninstalled",
            "java_version": "uninstalled",
            "last_used": 1624459875
        }],
        "computer_sid": "S-1-5-21-3284820969-3957662392-1842130629",
        "cpu_id": "561699bcbac7533644465b4f16637adf5870773b571f6d5b90bbfaeb2f0ba568",
        "device_identifier": "561699bcbac7533644465b4f16637adf5870773b571f6d5b90bbfaeb2f0ba568",
        "device_identifier_type": "hardware_serial",
        "device_name": "AMOSSPC",
        "disk_encryption_status": "Off",
        "domain_sid": "",
        "email": "",
        "epkey": "EPQC0C77F6MLXBCXCSWP",
        "firewall_status": "On",
        "hardware_uuid": "",
        "health_app_client_version": "2.14.0",
        "health_data_last_collected": 1624459875,
        "last_updated": 1624459875,
        "machine_guid": "6345e8c1-e717-4c68-a0f9-34cd0789e17f",
        "model": "",
        "os_build": "",
        "os_family": "Windows",
        "os_version": "10.0.19041.1052",
        "password_status": "Set",
        "type": "",
        "username": "amoss"
    },
}
```


Registered Devices

Retrieve Registered Devices

Returns a paged list of Duo Desktop registered devices. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset`. If no number or extension parameters are provided, the list will contain all registered devices. Requires "Grant resource - Read" API permission.

`GET` `/admin/v1/registered_devices`

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	Success.

RESPONSE FORMAT

Key	Value
`compkey`	Computer/Device key.
`device_name`	The endpoint's hostname.
`device_os`	Operating system on the device.
`identifier`	The unique device attribute value that identifies the endpoint.
`user`	Selected information about the end user attached to this device. See Retrieve Users for descriptions of the response fields.

EXAMPLE RESPONSE


```
{
  "metadata": {
    "total_objects": 2
},
"response": [
  {
    "compkey": "CRSFWW1YWVNUXMBJ1J29",
    "device_name": "CJONES-M-J1JF",
    "device_os": "Mac OS X",
    "identifier": "561699bcbac7533644465b4f16637adf5870773b571f6d5b90bbfaeb2f0ba568",
    "user": [
      {
        "alias1": null,
        "alias2": null,
        "alias3": null,
        "alias4": null,
        "aliases": {},
        "created": 1716491582,
        "email": "cjones@example.com",
        "enable_auto_prompt": true,
        "firstname": "Chris",
        "is_enrolled": true,
        "last_directory_sync": null,
```

```
      "last_login": 1716921006,
      "lastname": "Jones",
      "notes": "for CJ",
      "realname": "Chris Jones",
      "status": "active",
      "user_id":"DUO3MIXQG78V9A0C29GS",
      "username": "cjones"
    },
    {
      "alias1": null,
      "alias2": null,
      "alias3": null,
      "alias4": null,
      "aliases": {},
      "created": 1716562757,
      "email": "cjones@example.com",
      "enable_auto_prompt": true,
      "firstname": "Chris",
      "is_enrolled": true,
      "last_directory_sync": null,
      "last_login": 1716922823,
      "lastname": "Jones",
      "notes": "for CJ",
      "realname": "Chris Jones",
      "status": "active",
      "user_id": "DULM95N4M4RVYO2HRYDW",
      "username": "cjones"
    }
  ]
},
{
    "compkey": "CRSFWW1YWVNUXMBJ1J29",
    "device_name": "CJONES-M-J1JF",
    "device_os": "Mac OS X",
    "identifier": "561699bcbac7533644465b4f16637adf5870773b571f6d5b90bbfaeb2f0ba568",
    "user": [
      {
        "alias1": null,
        "alias2": null,
        "alias3": null,
        "alias4": null,
        "aliases": {},
        "created": 1716562757,
        "email": "cjones@example.com",
        "enable_auto_prompt": true,
        "firstname": "Chris",
        "is_enrolled": true,
        "last_directory_sync": null,
        "last_login": 1716922823,
        "lastname": "Jones",
        "notes": "for CJ",
        "realname": "Chris Jones",
        "status": "active",
        "user_id": "DULM95N4M4RVYO2HRYDW",
        "username": "cjones"
      }
    ]
  }
],
"stat": "OK"}
```


Delete Registered Devices

Deletes a single registered device corresponding to the `compkey` provided.

```
DELETE  /admin/v1/registered_devices/[registered_device_key]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No registered device was found with the given `registered_device_key`.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Retrieve Registered Devices by ID

Return the single registered device with `registered_device_key`. Requires "Grant resource - Read" API permission.

```
GET  /admin/v1/registered_devices/[registered_device_key]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No registered device was found with the given `registered_device_key`.

RESPONSE FORMAT

Refer to Retrieve Registered Devices for an explanation of the object's keys.

EXAMPLE RESPONSE

```
{
    "response": {
        "compkey": "CRSFWW1YWVNUXMBJ1J29",
        "device_name": "CJONES-M-J1JF",
        "device_os": "Mac OS X",
        "identifier": "561699bcbac7533644465b4f16637adf5870773b571f6d5b90bbfaeb2f0ba568",
        "user": [
            {
                "alias1": null,
                "alias2": null,
                "alias3": null,
```

```
                "alias4": null,
                "aliases": {},
                "created": 1717178659,
                "email": "cjones@example.com",
                "enable_auto_prompt": true,
                "firstname": "Chris",
                "is_enrolled": true,
                "last_directory_sync": null,
                "last_login": 1717441468,
                "lastname": "Jones",
                "notes": "for CJ",
                "realname": "Chris Jones",
                "status": "active",
                "user_id": "DUGE3VH8SOZP121FH3K6",
                "username": "cjones"
            }
        ]
    },
    "stat": "OK"
}
```


Passport

The Passport v2 API does not use the same authentication and request signing detailed earlier in this document. Our Python, Go, Java, and Node.JS API clients have been updated with the new authentication and signing requirements and include support for the Passport v2 API endpoint. We recommend you use the duo_client_python Python API client, the duo_api_golang Go API client, the duo_client_java Java API client, or the duo_api_nodejs Node.JS API client to interact with the Passport v2 endpoint.

Retrieve Passport Configuration

Returns the configuration for Duo Passport. Requires "Grant resource - Read" API permissions.

`GET` `/admin/v2/passport/config`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.

RESPONSE FORMAT

Key	Value
`disabled_groups`	List of groups that have Passport disabled. `group_id` — The group's ID. `group_name` — The group's name.
`enabled_groups`	List of groups that have Passport enabled. `group_id` — The group's ID.

`group_name`	The group's name.

`enabled_status`

The enabled status. One of:

Status	Value
`disabled`	Passport is disabled for all users.
`enabled`	Passport is enabled for all users.
`enabled-for-groups`	Passport is enabled for selected groups.
`enabled-with-exceptions`	Passport is enabled for all users except for selected groups.

EXAMPLE RESPONSE

```
{
    "response": {
        "disabled_groups": [],
        "enabled_groups": [
            {
                "group_id": "DAB12CDEFGHIJKLM3NOP",
                "group_name": "Acme Group"
            }
        ],
        "enabled_status": "enabled-for-groups"
    },
    "stat": "OK"
}
```

Modify Passport Configuration

Change the status and update the groups used for Duo Passport. Requires "Grant resource - Write" API permissions.

Note: This API call must be represented as JSON and cannot be passed in via URL parameters. Some Duo clients (Python or Java, for example) offer helper functions to handle requests to the Admin API.

`POST` `/admin/v2/passport/config`

PARAMETERS

Parameter	Required?	Description
`disabled_groups`	**Required**	List of groups that have Passport disabled. **Key** / **Value** `group_id` — The group's ID. `group_name` — The group's name.
`enabled_groups`	**Required**	List of groups that have Passport enabled. **Key** / **Value** `group_id` — The group's ID. `group_name` — The group's name.

`enabled_status`	**Required**	The enabled status. One of: **Status** / **Value** `disabled` — Passport is disabled for all users. `enabled` — Passport is enabled for all users. `enabled-for-groups` — Passport is enabled for selected groups. `enabled-with-exceptions` — Passport is enabled for all users except for selected groups.

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid or missing parameters.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "response": "",
  "stat": "OK"
}
```

Administrators

Retrieve Administrators

Returns a paged list of administrators. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. Requires "Grant administrators - Read" or "Grant administrators - Write" and "Grant resource - Read" API permissions.

```
GET /admin/v1/admins
```

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	Success.

RESPONSE FORMAT

Key	Value
`admin_id`	The administrator's ID.
`admin_units`	The list of administrative units (by `admin_unit_id`) to which the admin belongs. For an unrestricted admin, this is an empty list.
`created`	The administrator's creation date as a Unix timestamp. No creation date shown for administrators created before October 2021.
`email`	The administrator's email address.
`hardtoken`	Information about hardware tokens attached to the administrator, or `null` if none attached. See Retrieve Hardware Tokens for descriptions of the response values.
`last_directory_sync`	An integer indicating the last update to the administrator via directory sync as a Unix timestamp, or `null` if the administrator has never synced with an external directory or if the directory that originally created the user has been deleted from Duo.
`last_login`	An integer indicating the last time this administrator logged in, as a Unix timestamp, or `null` if the administrator has not logged in.
`name`	The administrator's full name.
`password_change_required`	Either `true` if the administrator must change their password at the next login, or `false` if no password change is required.
`phone`	The phone number in E.164 format.
`restricted_by_admin_units`	Is this administrator restricted by an administrative unit assignment? Either `true` or `false`. Must be set to `true` in order to add the admin to an administrative unit using the API.
`role`	The administrator's role. One of: "Owner", "Administrator", "Application Manager", "User Manager", "Security Analyst", "Help Desk", "Billing", "Phishing Manager", or "Read-only". Only present in the response if the customer edition includes the Administrative Roles feature.
`status`	The administrator account's status. One of: "Active" (admin can log in to Duo), "Disabled" (admin prevented from access), "Expired" (admin blocked from access due to inactivity), or "Pending Activation" (new admin must complete activation to gain access).
`webauthncredentials`	A list of WebAuthn authenticators that this administrator can use. **Key** / **Value** `credential_name` — Free-form label for the WebAuthn credential. `date_added` — The date the WebAuthn credential was registered in Duo. `webauthnkey` — The WebAuthn credential's registration identifier.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": [
    {
```

```
    "admin_id": "DEVKWVBJA7LO95OIBIS3",
    "admin_units": [],
     "created": 1648143942,
     "email": "rscott@example.com",
     "hardtoken": {
       "serial": "1234567890",
       "token_id": "DH1TIP00LBCIH887OPSV",
       "totp_step": null,
       "type": "d1"
     },
     "last_directory_sync": null,
     "last_login": 1343921403,
     "name": "Rachael Scott",
     "password_change_required": false,
     "phone": "+17345551000",
     "restricted_by_admin_units": false,
     "role": "Owner",
     "status": "Active",
     "webauthncredentials": [
       {
           "credential_name": "Security key",
           "date_added": 1679412977,
           "webauthnkey": "WA1HOM2JP00L6HLP5JYM"
       },
       {
           "credential_name": "Touch ID",
           "date_added": 1679497182,
           "webauthnkey": "WABNYB93OD4DVWVBC41X"
       }
     ]
   },
   {
     "admin_id": "DEVKWVBJA7P00LD4DIS3",
     "admin_units": [],
      "created": 1648146942,
      "email": "fulan@example.com",
      "hardtoken": {},
      "last_directory_sync": 1658850983,
      "last_login": 1343971403,
      "name": "Fulan Al Fulani",
      "password_change_required": false,
      "phone": "+17345551100",
      "restricted_by_admin_units": false,
      "role": "Administrator",
      "status": "Active"
    }
  ]
}
```


Create Administrator

Create a new administrator. Requires "Grant administrators - Write" API permission.

`POST` `/admin/v1/admins`

PARAMETERS

Parameter	Required?	Description
`email`	**Required**	Valid email address for the new administrator.
`name`	**Required**	Name for the new administrator.

`password`	Deprecated	Legacy parameter; ignored if specified. Formerly the password value for the new administrator.
`password_change_required`	Deprecated	This parameter may not be used when creating a new administrator, as the new admin does not have a password at creation.
`phone`	Optional	Phone number for the new administrator; E.164 format recommended (i.e. "+17345551212"). If no leading plus sign is provided then it is assumed to be a United States number and an implicit "+1" country code is prepended. Dashes and spaces are ignored. If this parameter is specified it cannot be empty.
`role`	Optional	The administrator's role. One of: "Owner", "Administrator", "Application Manager", "User Manager", "Security Analyst", "Help Desk", "Billing", "Phishing Manager", or "Read-only". The role names are case-sensitive. Defaults to "Owner" if not specified. Roles other than "Owner" are effective only if the customer edition includes the Administrative Roles feature.
`restricted_by_admin_units`	Optional	Is this administrator restricted by an administrative unit assignment? Either `true` or `false`. Defaults to `false` if not specified. Must be set to `true` in order to add the admin to an administrative unit using the API. Note that attempting to set to `true` for admins with the "Owner" role results in a failure response.
`send_email`	Optional	If set to `1`, the activation link and an introductory message will be emailed to the new administrator. If set to `0`, no email is sent, and the link is returned to the API method's caller only. Default: `0`.
`token_id`	Optional	The `token_id` of the hardware token to associate with the administrator.
`valid_days`	Optional	Number of days before the activation link expires. Default: `7` Maximum:: `31`

RESPONSE CODES

Response	Meaning
200	Success. Returns the newly created administrator.
400	Invalid or missing parameters, the role assigned may not be restricted by an administrative unit, or the provided email address is already in use by another administrator.

RESPONSE FORMAT

Returns the created single administrator object, with the same information as Retrieve Administrator by ID plus:

`activation_url_expires`	An integer indicating the timestamp of the activation link's expiration.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "activation_url": "https://admin-abcd1234.duosecurity.com/admins/activation/DAAR5SIVP00LYZA0W
    "activation_url_expires": 1604347975,
    "admin_id": "DEVKWVBJA7LO95OIBIS3",
    "admin_units": [],
    "created": 1648143942,
    "email": "rscott@example.com",
    "hardtoken": null,
    "last_directory_sync": null,
    "last_login": null,
```

```
    "name": "Rachael Scott",
    "password_change_required": false,
    "phone": null,
    "restricted_by_admin_units": false,
    "role": "Help Desk",
    "status": "Pending Activation"
  }
}
```


Retrieve Administrator by ID

Return the single administrator with the administrator ID `admin_id`. Requires "Grant administrators - Read" or "Grant administrators - Write" API permission.

```
GET  /admin/v1/admins/[admin_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No administrator was found with the given `admin_id`.

RESPONSE FORMAT

Returns the single administrator object, with the same information as Retrieve Administrators plus:

`activation_url`	Link to the activation form if an activation link exists for that admin.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
      "admin_id": "DEA0HJ95P00LNFXO9DEMK",
      "admin_units": [],
      "created": null,
      "email": "ellery.munson@example.com",
      "hardtoken": {
          "serial": "100048",
          "token_id": "DH89J6KJAF00BQ2ILS0K",
          "totp_step": null,
          "type": "h6"
      },
      "last_directory_sync": null,
      "last_login": 1679517329,
      "name": "Ellery Munson",
      "password_change_required": false,
      "phone": "+17345559542",
      "restricted_by_admin_units": false,
      "role": "Administrator",
      "status": "Active",
      "webauthncredentials": [
          {
              "credential_name": "Touch ID",
              "date_added": 1679517297,
```

```
                "webauthnkey": "WAJ3YT0D4DMSIW9F6MVC"
            }
        ]
    }
}
```

Modify Administrator

Change the name, phone number, or other properties of the administrator with the administrator ID `admin_id`. Requires "Grant administrators - Write" API permission.

`POST` `/admin/v1/admins/[admin_id]`

PARAMETERS

Parameter	Required?	Description
`name`	Optional	New name for the administrator. Read-only if the admin is managed by directory sync.
`phone`	Optional	The phone number; E.164 format recommended (i.e. "+17345551212"). If no leading plus sign is provided then it is assumed to be a United States number and an implicit "+1" country code is prepended. Dashes and spaces are ignored. If this parameter is specified it cannot be empty.
`password`	Deprecated	Legacy parameter; ignored if specified. Formerly the new password value for the administrator.
`password_change_required`	Optional	Specify `true` to require that the admin pick a new password at their next login, or `false` if no password change is required. May not be changed to `true` if the admin has external password management enabled.
`role`	Optional	New role for the administrator. One of: "Owner", "Administrator", "Application Manager", "User Manager", "Security Analyst", "Help Desk", "Billing", "Phishing Manager", or "Read-only". The role names are case-sensitive. Roles other than "Owner" are effective only if the customer edition includes the Administrative Roles feature. Read-only if the admin is managed by directory sync.
`restricted_by_admin_units`	Optional	Is this administrator restricted by an administrative unit assignment? Either `true` or `false`. Must be set to `true` in order to add the admin to an administrative unit using the API. Note that attempting to set to `true` for admins with the "Owner" role results in a failure response.
`status`	Optional	The desired administrator account status. Either "Active" or "Disabled" (case-sensitive). Administrators with the "Owner" role may not be disabled via API. To clear an inactive admin's "Expired" status, see Clear Administrator Expiration. Read-only if the admin is managed by directory sync.
`token_id`	Optional	The ID of the hardware token to associate with the administrator. Specify with no value to remove any existing token assignment for that administrator.

RESPONSE CODES

Response	Meaning
200	The administrator was modified successfully. The administrator object is also returned (see Retrieve Administrator by ID).
400	Invalid or missing parameters, or the role assigned may not be restricted by an administrative unit.
404	No administrator was found with the given `admin_id`.

RESPONSE FORMAT

Returns the single modified administrator object. Refer to Retrieve Administrators for an explanation of the object's keys.

EXAMPLE RESPONSE

Same as Retrieve Administrator by ID.

Delete Administrator

Delete the administrator with administrator ID `admin_id` from the system. Administrators managed by directory sync can not be deleted via API. Requires "Grant administrators - Write" API permission.

```
DELETE  /admin/v1/admins/[admin_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The administrator was deleted or did not exist.
400	The administrator could not be deleted.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Reset Administrator Authentication Attempts

Clear the number of failed login attempts for the administrator with `admin_id`. Re-enables an administrator who has been disabled due to too many failed authentication attempts. Requires "Grant administrators - Write" API permission.

```
POST  /admin/v1/admins/[admin_id]/reset
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The administrator's authentication failure count was set to zero.
404	No administrator was found with the given `admin_id`.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Clear Administrator Expiration

Clear expiration for the administrator with `admin_id` after the admin has been expired due to inactivity. The administrator's status changes from "Expired" to the status applied to that admin before inactive expiration, and restores access to the Duo Admin Panel if the effective status is "Active". Requires "Grant administrators - Write" API permission.

`POST` `/admin/v1/admins/[admin_id]/clear_inactivity`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The administrator's "Expired" status was cleared.
404	No administrator was found with the given `admin_id`.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Email Activation Link to Administrator Pending Activation

Email the current activation link to the administrator pending activation with the administrator ID `admin_id`. Requires "Grant administrators - Write" API permission.

`POST` `/admin/v1/admins/[admin_id]/activation_link/email`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	An email with the activation link was sent to the admin.
400	Invalid administrator for activation.
404	Invalid `admin_id`.

RESPONSE FORMAT

Key	Value
`admin_activation_id`	The ID of the administrator activation link.

`code`	Activation code used to create this activation link and message.
`email`	Email address of the administrator.
`expires`	An integer indicating the timestamp of the activation link's expiration.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": {
      "admin_activation_id": "DK9PHLB5Z8NZJRFSJX4Q",
      "code": "g793cfba2b4e8684164c6b8766baad08",
      "email": "rscott@example.com",
      "expires": 1604348536
  }
}
```

Delete Activation Link from Administrator Pending Activation

Deletes and invalidates the current activation link from the administrator pending activation with the administrator ID `admin_id`. Requires "Grant administrators - Write" API permission.

`DELETE` `/admin/v1/admins/[admin_id]/activation_link`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Any existing activation link was deleted and invalidated.
400	Invalid administrator for activation.
404	Invalid `admin_id`.

RESPONSE FORMAT

Same as Retrieve Administrator by ID.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": {
    "admin_id": "DEVKWVBJA7LO95OIBIS3",
    "admin_units": [],
    "created": 1648143942,
    "email": "rscott@example.com",
    "hardtoken": null,
    "last_directory_sync": null,
    "last_login": null,
    "name": "Rachael Scott",
    "password_change_required": false,
    "phone": null,
    "restricted_by_admin_units": false,
    "role": "Owner",
    "status": "Pending Activation"
```

```
    }
}
```

Create Activation Link for Administrator Pending Activation

Creates an activation link for the administrator pending activation with the administrator ID `admin_id`. There must not already be an existing activation link for the admin. Requires "Grant administrators - Write" API permission.

`POST` `/admin/v1/admins/[admin_id]/activation_link`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Creates an activation link for the admin.
400	Invalid administrator for activation or an activation link already exists for that admin.
404	Invalid `admin_id`.

RESPONSE FORMAT

Same as Retrieve Administrator by ID.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "activation_url": "https://admin-abcd1234.duosecurity.com/admins/activation/DAAR5SIVP00LYZA0W
    "admin_id": "DEVKWVBJA7LO95OIBIS3",
    "admin_units": [],
    "created": 1648143942,
    "email": "rscott@example.com",
    "hardtoken": null,
    "last_directory_sync": null,
    "last_login": null,
    "name": "Rachael Scott",
    "password_change_required": false,
    "phone": null,
    "restricted_by_admin_units": false,
    "role": "Owner",
    "status": "Pending Activation"
  }
}
```


Create Administrator Activation Link

Create a link to the activation form for a new administrator with email address `email`. The administrator will not actually be created until the activation form is completed with further information (like the administrator's name and phone number). Requires "Grant administrators - Write" API permission.

`POST` `/admin/v1/admins/activations`

PARAMETERS

Parameter	Required?	Description

`email`	**Required**	Email address for the new administrator. Must not already be in use by any other administrator or pending administrator activation.
`admin_name`	Optional	The full name of the administrator. The administrator's `email` will be used as the name if not specified.
`admin_role`	Optional	The administrator's role. One of: "Owner", "Administrator", "Application Manager", "User Manager","Security Analyst", "Help Desk", "Billing", "Phishing Manager", or "Read-only". The role names are case-sensitive. Defaults to "Owner" if not specified. Roles other than "Owner" are effective only if the customer edition includes the Administrative Roles feature.
`send_email`	Optional	If set to `1`, the activation link and an introductory message will be emailed to the new administrator. If set to `0`, no email is sent, and the link is returned to the API method's caller only. Default: `0`.
`valid_days`	Optional	Number of days before the link expires. Default: `7` Maximum: `31`

RESPONSE CODES

Response	Meaning
200	Activation link is returned (and optionally emailed).
400	Invalid or missing parameters, or `email` is already in use by an existing administrator or pending activation request.

RESPONSE FORMAT

Key	Value
`admin_activation_id`	The ID of the administrator activation link.
`admin_role`	The administrator role assigned to the new admin.
`code`	Activation code used to create this activation link and message.
`email`	Email address supplied by the caller. If the activation form is completed a new administrator will be created with this email address.
`email_sent`	`1` if the introductory message was emailed to the new administrator; `0` otherwise.
`expires`	An integer indicating the Unix timestamp of the activation link's expiration.
`link`	Link to the activation form.
`message`	Introductory message body.
`subject`	Introductory message subject.
`valid_days`	An integer indicating the number of days before the activation link expires.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "admin_activation_id": "DK9PHLB5Z8NZJRFSJX4Q",
    "admin_name": "Soren Ogilby",
    "admin_role": "Read-only",
    "code": "g793cfba2b4e8684164c6b8766baad08",
    "email": "sogilby@example.com",
    "email_sent": 1,
    "expires": 1550075404,
    "link": "https://admin-abcd1234.duosecurity.com/activation/g793cfba2b4e8684164c6b8766baad08",
    "message": "\nHello sogilby@example.com,\n\nYou have been added as a Duo administrator for th
    "subject": "Set up your administrator account on Duo Security",
    "valid_days": 7
```

```
  },
}
```

Retrieve Pending Administrator Activations

Returns a paged list of pending administrator activations. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. Requires "Grant administrators - Read" or "Grant administrators - Write" API permission.

```
GET  /admin/v1/admins/activations
```

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	A list of pending admin activations is returned.
400	Invalid paging parameters.

RESPONSE FORMAT

Key	Value
`admin_activation_id`	The ID of the administrator activation link.
`code`	Activation code used to create this activation link and message.
`email`	Email address supplied by the caller. If the activation form is completed a new administrator will be created with this email address.
`expires`	An integer indicating the timestamp of the activation link's expiration.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": {
    "admin_activation_id": "DK9PHLB5Z8NZJRFSJX4Q",
    "code": "g793cfba2b4e8684164c6b8766baad08",
    "email": "sogilby@example.com",
    "expires": 1550075404
  },
}
```

Delete Pending Administrator Activation

Delete the pending admin activation with ID `admin_activation_id` from the system. Requires "Grant administrators - Write" API permission.

```
DELETE  /admin/v1/admins/activations/[admin_activation_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The pending admin activation link was deleted or did not exist.
404	Invalid `admin_activation_id` format.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Synchronize Admin from Directory

Initiate a sync to create, update, or mark for deletion the administrator specified by `email` against the directory specified by the `directory_key`. The `directory_key` for a directory can be found by navigating to **Users** → **Administrators** → **Admin Directory Sync** in the Duo Admin Panel, and then clicking on the configured directory. Learn more about syncing individual admins from Active Directory, OpenLDAP, or Entra ID. Requires "Grant administrators - Write" API permission.

```
POST  /admin/v1/admins/directorysync/[directory_key]/syncadmin
```

PARAMETERS

Parameter	Required?	Description
`email`	Required	Email address of the admin to update or create via directory sync. This should be the same as the value for the admin's email attribute in the source directory as configured in the sync. Administrators with "Owner" role may not be synced.

RESPONSE CODES

Response	Meaning
200	The admin was synced successfully and updated or added in Duo. The admin object is also returned (see Retrieve Users).
404	The specified `email` or `directory_key` was incorrect, the admin is not managed by the specified directory, or the admin is not a member of any source directory group specified in the sync configuration.
429	Too many requests; try again later.

RESPONSE FORMAT

Returns the single synced administrator object with an additional `message` stating the admin synced successfully. Refer to Retrieve Administrators for an explanation of the object's keys.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response":{
    "admin": {
        "admin_id": "DE98U1VAFZ6CO4T06HZO",
        "admin_units": [],
        "created": 1658850929,
        "email": "sogilby@example.com",
        "hardtoken": null,
        "last_directory_sync": 1658850983,
        "last_login": null,
        "name": "Soren Ogilby",
        "password_change_required": false,
        "phone": "+17345559777",
        "restricted_by_admin_units": false,
        "role": "Help Desk",
        "status": "Pending Activation"
    },
    "message": "User sogilby@example.com synced successfully."
}
```


Retrieve Admin External Password Management Status

Returns a paged list of administrators indicating whether they have been configured for external password management. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. Requires "Grant administrators - Read" or "Grant administrators - Write" API permission.

`GET` `/admin/v1/admins/password_mgmt`

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100` ; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

This API endpoint has no additional parameters.

RESPONSE CODES

Response	Meaning
200	Returns a list of administrators with each admin's external password management status.
400	Invalid paging parameters.

RESPONSE FORMAT

Key	Value

`admin_id`	The administrator's ID.
`email`	The administrator's email address.
`has_external_password_mgmt`	Either `true` if the administrator's password may be set via API, or `false` if passwords are self-managed (default).

EXAMPLE RESPONSE

```
{
   "stat": "OK",
    "response": [
        {
            "admin_id": "DEORP00LUIXPGD4DCC37",
            "email": "sogilby@example.com",
            "has_external_password_mgmt": false
        },
    ],
}
```

Retrieve Admin External Password Management Status by ID

Returns the external password management configuration for the single administrator with the administrator ID `admin_id`.
Requires "Grant administrators - Read" or "Grant administrators - Write" API permission.

`GET` `/admin/v1/admins/[admin_id]/password_mgmt`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Returns the specified administrator's password management status.
404	No administrator was found with the given `admin_id`.

RESPONSE FORMAT

Same as Retrieve Admin External Password Management Status.

EXAMPLE RESPONSE

```
{
    "stat": "OK",
    "response": {
        "admin_id": "DEORP00LUIXPGD4DCC37",
        "email": "sogilby@example.com",
        "has_external_password_mgmt": false
    },
}
```

Modify Admin External Password Management Status or Password

Enable or disable an administrator, specified by `admin_id`, for external password management, or set the password for an administrator with `has_external_password_mgmt` set to `true` (either passed in with the same POST or previously set).

Administrator passwords must have at least twelve characters, and may also require a mix of character types depending on your Admin Password Policy settings. New passwords will be checked against common passwords, usernames, and other account information to ensure uniqueness.

Setting `has_external_password_mgmt` also updates the administrator account's `password_change_required` value. When `has_external_password_mgmt` is set to `true`, `password_change_required` is updated to `false`, as enabling external password management restricts administrators from performing self-service password resets via the Duo Admin Panel UI. When `has_external_password_mgmt` is set to `false`, `password_change_required` is updated to `true` to ensure that an administrator no longer subject to external password management updates their password to a new value not known by the external system.

```
POST /admin/v1/admins/[admin_id]/password_mgmt
```

PARAMETERS

Parameter	Required?	Description
`has_external_password_mgmt`	Optional	Specify `true` if the administrator's password may be set via API, or `false` if passwords are self-managed. If specifying `true` you may also send a `password` value in the same operation.
`password`	Optional	New password for the administrator. May be sent in the same operation with `has_external_password_mgmt=true`

RESPONSE CODES

Response	Meaning
200	Success.
400	Password specified when external password management not enabled for the admin, or new password does not satisfy the password policy.
404	No administrator was found with the given `admin_id`.

RESPONSE FORMAT

Key	Value
`admin_id`	The administrator's ID.
`email`	The administrator's email address.
`has_external_password_mgmt`	Returns `true` if the administrator's password may be set via API, or `false` if passwords are self-managed. If the setting was changed in the request then the new value is returned.
`password_changed`	Returns `true` if the administrator's password was changed in the request, or `false` if the request did not attempt to change the password.

EXAMPLE RESPONSE

```
{
    "stat": "OK",
    "response": {
            "admin_id": "DEORP00LUIXPGD4DCC37",
            "email": "sogilby@example.com",
            "has_external_password_mgmt": true,
            "password_changed": true
    },
}
```

Retrieve Administrator Authentication Factors

Retrieve a list of the secondary authentication methods permitted for administrator log on to the Duo Admin Panel. Requires "Grant administrators - Read" or "Grant administrators - Write" API permission.

```
GET /admin/v1/admins/allowed_auth_methods
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.

RESPONSE FORMAT

Key	Value
`hardware_token_enabled`	If `true`, administrators may authenticate to the Duo Admin Panel with an OTP hardware token. If `false`, administrators may not use OTP hardware tokens.
`mobile_otp_enabled`	If `true`, administrators may authenticate to the Duo Admin Panel with a passcode generated by the Duo Mobile app. If `false`, administrators may not use Duo Mobile passcodes.
`push_enabled`	If `true`, administrators may authenticate to the Duo Admin Panel by approving a push request in the Duo Mobile app. If `false`, administrators may not approve login with Duo Push.
`sms_enabled`	If `true`, administrators may authenticate to the Duo Admin Panel with a passcode received via SMS. If `false`, administrators may not use SMS passcodes.
`verified_push_enabled`	If `true`, administrators may authenticate to the Duo Admin Panel with a Verified Duo Push. If `false`, administrators are not required to enter a verification code in the Duo Mobile app to approve a login request.
`verified_push_length`	The number of digits a Verified Duo Push requires the admin to enter in the Duo Mobile app to approve a login request. An integer between `3` and `6`, inclusive. Defaults to `3`. Is `null` if `verified_push_enabled` is `false`.
`voice_enabled`	If `true`, administrators may authenticate to the Duo Admin Panel by approving the request received via phone call. If `false`, administrators may not approve login with a phone call.
`webauthn_enabled`	If `true`, administrators may authenticate to the Duo Admin Panel with a WebAuthn credential (also known as a passkey). If `false`, administrators may not use passkeys.
`yubikey_enabled`	If `true`, administrators may authenticate to the Duo Admin Panel with a Yubikey token. If `false`, administrators may not use Yubikey tokens.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": {
    "hardware_token_enabled": true,
    "mobile_otp_enabled": true,
    "push_enabled": true,
    "sms_enabled": false,
    "verified_push_enabled": false,
    "verified_push_length": null,
    "voice_enabled": false,
    "webauthn_enabled": true,
    "yubikey_enabled": true
  }
}
```

Restrict Administrator Authentication Factors

Enable or disable secondary authentication methods permitted for administrator log on to the Duo Admin Panel. When no methods are restricted Duo administrators may use any available two-factor method. Any method not explicitly set to true in the POST is disabled. Requires "Grant administrators - Write" API permission.

`POST` `/admin/v1/admins/allowed_auth_methods`

PARAMETERS

Parameter	Required?	Description
`hardware_token_enabled`	**Optional, but at least one valid factor must be set to true.**	If `true`, administrators may authenticate to the Duo Admin Panel with an OTP hardware token. If `false` or not specified, administrators may not use OTP hardware tokens.
`mobile_otp_enabled`	**Optional, but at least one valid factor must be set to true.**	If `true`, administrators may authenticate to the Duo Admin Panel with a passcode generated by the Duo Mobile app. If `false` or not specified, administrators may not use Duo Mobile passcodes.
`push_enabled`	**Optional, but at least one valid factor must be set to true.**	If `true`, administrators may authenticate to the Duo Admin Panel by approving a push request in the Duo Mobile app. If `false` or not specified, administrators may not approve login with Duo Push.
`sms_enabled`	**Optional, but at least one valid factor must set to true.**	If `true`, administrators may authenticate to the Duo Admin Panel with a passcode received via SMS. If `false` or not specified, administrators may not use SMS passcodes.
`verified_push_enabled`	**Optional, but at least one valid factor must set to true.**	If `true`, administrators may authenticate to the Duo Admin Panel with a Verified Duo Push. If `false`, administrators are not required to enter a verification code in the Duo Mobile app to approve a login request.
`verified_push_length`	**Optional, but `verified_push_enabled` must be `true`.**	The number of digits a Verified Duo Push requires the admin to enter in the Duo Mobile app to approve a login request. An integer between `3` and `6`, inclusive. Defaults to `3` if not specified.
`voice_enabled`	**Optional, but at least one valid factor must be set to true.**	If `true`, administrators may authenticate to the Duo Admin Panel by approving the request received via phone call. If `false` or not specified, administrators may not approve login with a phone call.
`webauthn_enabled`	**Optional, but at least one valid factor must be set to true.**	If `true`, administrators may authenticate to the Duo Admin Panel with a WebAuthn credential (also known as a passkey). If `false` or not specified, administrators may not use passkeys.
`yubikey_enabled`	**Optional, but at least one valid factor must be set to true.**	If `true`, administrators may authenticate to the Duo Admin Panel with a Yubikey token. If `false` or not specified, administrators may not use Yubikey tokens.

RESPONSE CODES

Response	Meaning
200	The settings were modified successfully. The settings object is also returned (see Retrieve Administrator Authentication Factors).
400	Invalid or missing parameters. For example, no valid factor was specified.

RESPONSE FORMAT

Same as Retrieve Administrator Authentication Factors

EXAMPLE RESPONSE

Same as Retrieve Administrator Authentication Factors

Administrative Units

Retrieve Administrative Units

Returns a paged list of all administrative units if no parameters specified. To fetch all results, call repeatedly with the `offset` parameter as long as the result metadata has a `next_offset` value. Requires "Grant administrators - Read" or "Grant administrators - Write" API permission.

Optionally specify at most one parameter to return a list of administrative units associated with the given administrator, group, or integration.

`GET` `/admin/v1/administrative_units`

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `100`; Max: `500`
`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

Parameter	Required?	Description
`admin_id`	Optional	A Duo administrator's `admin_id`.
`group_id`	Optional	A Duo group's `group_id`.
`integration_key`	Optional	The `integration_key` for a Duo integration.

RESPONSE CODES

Response	Meaning
200	Success.
404	No administrative unit was found with the given `admin_id`, `group_id`, or `integration_key`.

RESPONSE FORMAT

Key	Value
`admin_unit_id`	The administrative unit's unique ID.
`description`	The administrative unit's description.
`name`	The administrative unit's name.
`restrict_by_groups`	Does the administrative unit specify groups? Either `true` or `false`.
`restrict_by_integrations`	Does the administrative unit specify integrations? Either `true` or `false`.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": [{
    "admin_unit_id": "AUT0HJ753KH67HGF4S7H",
    "description": "Acme Corp Europe, Middle East, and Africa",
    "name": "Acme EMEA",
    "restrict_by_groups": true,
    "restrict_by_integrations": true
    },
    {
    "admin_unit_id": "AUDJ753KH6Z252X1B2B4",
    "description": "Acme Corp United States",
```

```
    "name": "Acme USA",
    "restrict_by_groups": true,
    "restrict_by_integrations": true
  }]
}
```


Retrieve Administrative Unit Details

Returns details for a single administrative unit with `admin_unit_id`. Requires "Grant administrators - Read" or "Grant administrators - Write" API permission.

```
GET  /admin/v1/administrative_units/[admin_unit_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.
404	No administrative unit was found with the given `admin_unit_id`.

RESPONSE FORMAT

Key	Value
`admin_unit_id`	The administrative unit's unique ID.
`admins`	The administrators assigned to the new administrative unit, listed by `admin_id`.
`description`	The administrative unit's description.
`groups`	The groups assigned to the new administrative unit, listed by `group_id`.
`integrations`	The integrations assigned to the new administrative unit, listed by `integration_key`.
`name`	The administrative unit's name.
`restrict_by_groups`	Does the administrative unit specify groups? Either `true` or `false`.
`restrict_by_integrations`	Does the administrative unit specify integrations? Either `true` or `false`.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": [{
    "admin_unit_id": "AUTU3AA76HG76K9GPFJ2",
    "admins": [
      "DEA76HG76K45TQHBMFI4"
    ],
    "description": "Acme Networking and VPN Admins",
    "groups": [],
    "integrations": [
      "DIBA76HEQMHRWA76KSSH"
    ],
    "name": "Acme Net Admins",
    "restrict_by_groups": false,
    "restrict_by_integrations": true
  }
}
```

Add Administrative Unit

Add a new administrative unit with specified administrators, groups, or other parameters. Requires "Grant administrators - Write" API permission.

```
POST  /admin/v1/administrative_units
```

PARAMETERS

Parameter	Required?	Description
`name`	Required	The name of the new administrative unit. Must be unique amongst all administrative units.
`description`	Required	A description of the new administrative unit.
`restrict_by_groups`	Required	Does the new administrative unit specify groups? Default: `false`.
`restrict_by_integrations`	Optional	Does the new administrative unit specify integrations? Default: `false`.
`admins`	Optional	One or more `admin_id` values to assign administrators to the new administrative unit. The administrator user must have `restricted_by_admin_units` set to `true` before attempting to assign them to an administrative unit via the API.
`groups`	Optional	One or more `group_id` values to assign groups to the new administrative unit.
`integrations`	Optional	One or more `integration_key` values to assign integrations to the new administrative unit.

RESPONSE CODES

Response	Meaning
200	The administrative unit was created. The newly created unit is also returned.
400	Invalid or missing parameter(s), or administrative unit already exists with the given `name`.

RESPONSE FORMAT

Same as Retrieve Administrative Unit Details.

Modify Administrative Unit

Change the name, description, assigned administrators, groups, and/or integrations of the administrative unit with `admin_unit_id`. Requires "Grant administrators - Write" API permission.

```
POST  /admin/v1/administrative_units/[admin_unit_id]
```

PARAMETERS

Parameter	Required?	Description
`name`	Optional	The new name of the administrative unit. Must be unique amongst all administrative units.
`description`	Optional	An updated description of the administrative unit.
`restrict_by_groups`	Optional	Change whether the administrative unit specifies groups. Default: `false`.

`restrict_by_integrations`	Optional	Change whether the administrative unit specifies integrations. Default: `false`.
`admins`	Optional	One or more `admin_id` values to assign additional administrators to the administrative unit. The administrator user must have `restricted_by_admin_units` set to `true` before attempting to assign them to an administrative unit via the API.
`groups`	Optional	One or more `group_id` values to assign additional groups to the administrative unit.
`integrations`	Optional	One or more `integration_key` values to assign additional integrations to the administrative unit.

RESPONSE CODES

Response	Meaning
200	The administrative unit was modified. Also returns unit details.
400	Invalid parameter(s), or an administrative unit with the specified `admin_unit_id` does not exist.

RESPONSE FORMAT

Same as Retrieve Administrative Unit Details.

Add Administrator to Administrative Unit

Assign the administrator with `admin_id` to the administrative unit with `admin_unit_id`. The administrator user must have `restricted_by_admin_units` set to `true` before attempting to assign them to an administrative unit via the API. Requires "Grant administrators - Write" API permission.

```
POST /admin/v1/administrative_units/[admin_unit_id]/admin/[admin_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The administrative unit was modified. Also returns unit details.
400	Invalid `admin_unit_id` or `admin_id`, or the `restricted_by_admin_units` value is `false` for that administrator.

RESPONSE FORMAT

Same as Retrieve Administrative Unit Details.

Remove Administrator from Administrative Unit

Unassign the administrator with `admin_id` from the administrative unit with `admin_unit_id`. The administrator user will still have `restricted_by_admin_units` set to `true`, and if the admin is not assigned to any other admin unit they will not be able to view any users or integrations. Be sure to change the value of `restricted_by_admin_units` to `false` to permit that admin to view all users and integrations. Requires "Grant administrators - Write" API permission.

```
DELETE /admin/v1/administrative_units/[admin_unit_id]/admin/[admin_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The administrative unit was modified. Also returns unit details.
400	Invalid `admin_unit_id` or `admin_id`.

RESPONSE FORMAT

Same as Retrieve Administrative Unit Details.

Add Group to Administrative Unit

Assign the group with `group_id` to the administrative unit with `admin_unit_id`. Requires "Grant administrators - Write" API permission.

`POST` `/admin/v1/administrative_units/[admin_unit_id]/group/[group_id]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The administrative unit was modified. Also returns unit details.
400	Invalid `admin_unit_id` or `group_id`.

RESPONSE FORMAT

Same as Retrieve Administrative Unit Details.

Remove Group from Administrative Unit

Unassign the group with `group_id` from the administrative unit with `admin_unit_id`. Requires "Grant administrators - Write" API permission.

`DELETE` `/admin/v1/administrative_units/[admin_unit_id]/group/[group_id]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The administrative unit was modified. Also returns unit details.
400	Invalid `admin_unit_id` or `group_id`.

RESPONSE FORMAT

Same as Retrieve Administrative Unit Details.

Add Integration to Administrative Unit

Assign the integration with `integration_key` to the administrative unit with `admin_unit_id`. Requires "Grant administrators - Write" API permission.

`POST` `/admin/v1/administrative_units/[admin_unit_id]/integration/[integration_key]`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The administrative unit was modified. Also returns unit details.
400	Invalid `admin_unit_id` or `integration_key`.

RESPONSE FORMAT

Same as Retrieve Administrative Unit Details.

Remove Integration from Administrative Unit

Unassign the integration with `integration_key` from the administrative unit with `admin_unit_id`. Requires "Grant administrators - Write" API permission.

```
DELETE  /admin/v1/administrative_units/[admin_unit_id]/integration/[integration_key]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The administrative unit was modified. Also returns unit details.
400	Invalid `admin_unit_id` or `integration_key`.

RESPONSE FORMAT

Same as Retrieve Administrative Unit Details.

Delete Administrative Unit

Delete the administrative unit with `admin_unit_id` from the system. Requires "Grant administrators - Write" API permission.

```
DELETE  /admin/v1/administrative_units/[admin_unit_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The administrative unit was deleted or did not exist.

RESPONSE FORMAT

Empty string.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": ""
}
```

Logs

Authentication Logs

Returns a paged list of authentication log events ranging from the last 180 days up to as recently as two minutes before the API request. To fetch all results, call repeatedly with the `next_offset` paging parameter as long as the result metadata has `next_offset` values. Requires "Grant read log" API permission.

There is an intentional two minute delay in availability of new authentications in the API response. Duo operates a large scale distributed system, and this two minute buffer period ensures that calls will return consistent results. Querying for results more recent than two minutes will return as empty.

We recommend requesting logs no more than once per minute.

The v2 handler provides new filtering and querying capabilities unavailable in the legacy v1 handler. This includes the ability to filter on users, groups, applications, authentication results, factors, and time ranges.

`GET or POST` `/admin/v2/logs/authentication`

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Allow Multiple?	Description
`limit`	Optional	No	The maximum number of records returned. Default: `100` ; Max: `1000`
`next_offset`	Optional	Yes	The offset at which to start record retrieval. This value is provided in the metadata in the form of a 13 character date string in milliseconds and the event `txid` . Both of these values must be provided when used, separated by a comma. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: None Example: `next_offset=1547486297000,5bea1c1e-612c-4f1d-b310-75fd31385b15`
`sort`	Optional	No	The order in which to return records. One of: `ts:asc` — Return logs in chronological order. `ts:desc` — Return logs in reverse chronological order.

Parameter	Required?	Allow Multiple?	Description
`mintime`	**Required**	No	Return records that have a 13 character Unix `timestamp` in milliseconds of `mintime` or later. This value must be strictly less than `maxtime` .

			Example: `1661022959934`
`maxtime`	**Required**	No	Return records that have a 13 character Unix `timestamp` in milliseconds of `maxtime` or earlier. This value must be strictly greater than `mintime`. Example: `1661022969934`
`applications`	Optional	Yes. Multiple values create an `OR` query.	An integration's `integration_key` or the `key` value for an application returned in the authentication log output. Default: Return logs for all applications.
`users`	Optional	Yes. Multiple values create an `OR` query.	A user's `user_id` or the `key` value for a user returned in the authenticatio log output. Default: Return logs for all users.
`assessment`	Optional	Yes. Multiple values create an `OR` query.	The risk-based authentication assessment based on risk-based factor select (RBFS) and risk-based remembered device (RBRD) policy enforcement. This information is available to Duo Premier and Duo Advantage plan customers. One of:

Value	Description
`factors restricted`	(Deprecated) Return events where the authenticati was stepped up due to low RBFS trust.
`New device session`	Return events where an RBRD session was just created.
`New session required`	Return events where an RBRD session was detecte but the session cookie was tampered.
`No detections`	Return events where the authentication had normal RBFS trust with a not enabled RBRD policy.
`No enforcement`	Return events where risk was detected but step-up was suppressed due to earlier activity justifying it as low risk.
`normal`	(Deprecated) Return events where the authenticati was not stepped up and a RBRD policy was not applied.
`policy not applied`	(Deprecated) Return events where the authenticati did not have a RBFS or RBRD policy applied.
`re-auth required`	(Deprecated) Return events where re-authenticatio was required due to low RBRD trust and was not affected by an RBFS policy.
`remembered device`	(Deprecated) Return events where the authenticati had normal RBRD trust and was not affected by an RBFS policy.
`Risk-based policy not enabled`	Return events where RBFS was not enabled, but lo trust was detected.
`Session verified`	Return events where a valid RBRD session was detected.
`Step-up cleared`	Return events where the authentication was steppe up due to earlier activity, but cleared by a more secure factor.
`Step-up initiated`	Return events where the authentication caused subsequent authentications to be stepped-up due t low RBFS trust.
`Step-up initiated & cleared`	Return events where the authentication was steppe up due to low RBFS trust, but instantly cleared by a more secure factor.

			`User stepped up` — Return events where the authentication was steppe up due to earlier activity. Default: Return logs for all assessments.
`detections`	Optional	Yes. Multiple values create an `OR` query.	The risk-based authentication detections found during or after an authentica attempt. This information is available to Duo Premier and Duo Advantage plan customers. One of: "consecutive failures", "country code mismatch", "credential stuffing", "device distance", "expired cookie", "manual removal by the user", "novel asn "novel ip and wifi fingerprint", "policy change", "previously marked fraud", "pu harassment", "service outage detected", "session not found", "tampered cookie", or "unrealistic travel". Default: Return logs for all detections.
`event_types`	Optional	Yes. Multiple values create an `OR` query.	The type of authentication event. One of: `authentication` — Return events for authentication attempts. `enrollment` — Return events related to a user completing Duo's inline enrollment.
`factors`	Optional	Yes. Multiple values create an `OR` query.	The factor or method used for an authentication attempt. One of: `bypass_code` — Return events where the authenticati factor was a bypass code. `Desktop Authenticator` — Return events where the authenticati factor was a desktop authenticator. `digipass_go_7_token` — Return events where the authenticati factor was a Digipass GO 7 token purchased from Duo. `duo_mobile_passcode` — Return events where the authenticati factor was a HOTP or TOTP passcoc generated by Duo Mobile. `duo_mobile_passcode_hotp` — Return events where the authenticati factor was a HOTP passcode generated by Duo Mobile. `duo_mobile_passcode_totp` — Return events where the authenticati factor was a TOTP passcode generated by Duo Mobile. `duo_push` — Return events where the authenticati factor was "Duo Push". `hardware_token` — Return events where the authenticati factor was a hardware token passcoc `not_available` — Return events where the authenticati factor is not available. `passcode` — Return events where the authenticati factor was a passcode not identified another known type. `phone_call` — Return events where the authenticati factor was a phone call. `Platform authenticator (2fa)` — Return events where the authenticati factor was a platform authenticator. `remembered_device` — Return events where the authenticati factor was the remembered device

			token from a previous authentication success.
			`Roaming authenticator (2fa)` Return events where the authenticati factor was a roaming authenticator.
			`sms_passcode` Return events where the authenticati factor was an SMS passcode.
			`sms_refresh` Return events where the user requested a refresh batch of SMS passcodes.
			`trusted_mobile_authenticator` Return events where the effective authentication factor Duo Mobile Inlin Auth on an Android or iOS device.
			`trusted_network` Return events where the effective authentication factor was an authoriz network.
			`u2f_token` Return events where the authenticati factor was a U2F token.
			`verified_duo_push` Return events where the authenticati factor was "Verified Duo Push".
			`WebAuthn Chrome Touch ID` Return events where the authenticati factor was Apple Touch ID with the Chrome browser. This property will b deprecated in a future release and b replaced with `Platform authenticator (2fa)`.
			`WebAuthn Credential` Return events where the authenticati factor was a WebAuthn authenticator other than a security key or Touch ID
			`WebAuthn Security Key` Return events where the authenticati factor was a FIDO2 security key. This property will be deprecated in a futur release and be replaced with `Roami authenticator (2fa)`.
			`yubikey_code` Return events where the authenticati factor was a Yubikey OTP token passcode.
`formatter`	Optional	No	This parameter is currently in Public Preview. Specifies the log format. If omitted, logs are returned in the default format. Or of: **Value** / **Description** `0` Returns logs in the default structure shown in this documentation. `1` Returns logs in Open Cybersecurity Schema Framework (OCSF) Authentication class format (v1.2). To learn more about OCSF, pleas refer to the official OCSF documentation .
`groups`	Optional	Yes. Multiple values create an `OR` query.	A group's `group_id` or the `key` value for a group returned in the authentication log output. Default: Return logs for all groups.
`phone_numbers`	Optional	Yes. Multiple values create an `OR` query.	A phone's `number` as returned in the authentication log output. If the phone has been given a text `name` then both are returned in the format `name (number)`. Default: Return logs for all phone numbers used.

`reasons`	Optional	Yes. Multiple values create an `OR` query.	The reason associated with an authentication attempt. One of: **Value** / **Descriptio** `user_marked_fraud` / Return even where authenticatic was denied because the end user explicitly marked "fraudulent". `deny_unenrolled_user` / Return even where authenticatic was denied because of t following policy: "deny not enrolled users". `error` / Return even where authenticatic was denied because of a error. `locked_out` / Return even generated by users that ar locked out. `user_disabled` / Return even where authenticatic was denied because the user was disabled. `user_cancelled` / Return even where authenticatic was denied because the end user cancelled the request. `invalid_passcode` / Return even where authenticatic was denied because the passcode wa invalid. `no_response` / Return even where authenticatic was denied because the was no response fro the user. `no_keys_pressed` / Return even where authenticatic was denied because no keys were pressed to accept the auth. `call_timed_out` / Return even where authenticatic was denied because the call was not answered or call

	authenticatic timed out for an indeterminat reason.
`location_restricted`	Return even where authenticatic was denied because the end user's location was restricted.
`factor_restricted`	Return even where authenticatic was denied because the authenticatic method usec was not allowed.
`platform_restricted`	Return even where authenticatic was denied because the access platform was not allowed.
`version_restricted`	Return even where authenticatic was denied because the software version was not allowed.
`rooted_device`	Return even where authenticatic was denied because the approval device was rooted.
`no_screen_lock`	Return even where authenticatic was denied because the approval device does not have screen lock enabled.
`touch_id_disabled`	Return even where authenticatic was denied because the approval device's biometrics (fingerprint, Face ID or Touch ID) is disabled.
`no_disk_encryption`	Return even where authenticatic was denied because the approval device did nc have disk encryption enabled.
`anonymous_ip`	Return even where authenticatic

	was denied because the authenticatic request cam from an anonymous address.
`out_of_date`	Return even where authenticatic was denied because the software was out of date.
`denied_by_policy`	Return even where authenticatic was denied because of a policy.
`software_restricted`	Return even where authenticatic was denied because of software restriction.
`no_duo_certificate_present`	Return even where authenticatic was denied because the was no Duo certificate present.
`user_provided_invalid_certificate`	Return even where authenticatic was denied because an invalid managemen certificate wa provided.
`could_not_determine_if_endpoint_was_trusted`	Return even where authenticatic was denied because it could not be determined i the endpoint was trusted.
`frequent_attempts`	Return even where authenticatic was denied because of frequent attempts.
`invalid_management_certificate_collection_state`	Return even where authenticatic was denied because of a invalid managemen certificate collection state.
`no_referring_hostname_provided`	Return even where authenticatic was denied because no referring hostname wa provided.
`invalid_referring_hostname_provided`	Return even where

	authenticatic was denied because an invalid referring hostname wa provided.
`no_web_referer_match`	Return even where authenticatic was denied because an invalid referring hostname di not match ar application's hostnames li
`endpoint_failed_google_verification`	Return even where authenticatic was denied because the endpoint fail Google verification.
`endpoint_is_not_trusted`	Return even where authenticatic was denied because the endpoint wa not trusted.
`invalid_device`	Return even where authenticatic was denied because the device was invalid.
`endpoint_is_not_in_management_system`	Return even where authenticatic was denied because the endpoint is n in a managemen system.
`no_activated_duo_mobile_account`	Return even where authenticatic was denied because the end user doe not have an activated Du Mobile app account.
`queued_inflight_auth_expired`	Return even where authenticatic was denied when more authenticatic than the number allowed by th lockout threshold are started simultaneous The authenticatic past the threshold are queued, and then remove from the que after enough failures trigg a lockout.

`allow_unenrolled_user`	Return even where authenticatic was success because of t following policy: "allow not enrolled users".
`bypass_user`	Return even where authenticatic was success because a bypass code was used.
`trusted_network`	Return even where authenticatic was success because the end user wa on a trusted network.
`remembered_device`	Return even where authenticatic was success because the end user wa on a remembered device.
`trusted_location`	Return even where authenticatic was success because the end user wa in a trusted location.
`user_approved`	Return even where authenticatic was success because the end user approved the authenticatic request.
`valid_passcode`	Return even where authenticatic was success because the end user use a valid passcode.
`allowed_by_policy`	Return even where authenticatic was success because of a policy.
`allow_unenrolled_user_on_trusted_network`	Return even where authenticatic was success because the unenrolled user's acces device was c an authorize network.
`user_not_in_permitted_group`	Return even where authenticatic was denied because the user did not

				belong to on of the Permitted Groups specified in t application's settings.
			`verification_code_correct`	Return even where authenticatic was success because of a Verified Duo Push.
			`verification_code_missing`	Return even where authenticatic was denied because the user used ar old version c Duo Mobile that does no support Verified Duo Push.
			`verification_code_incorrect`	Return even where authenticatic was denied because the user entered the wrong code when approving a Verified Duo Push.

Default: Return logs for any result. Filtering on all values is equivalent to the default.

Note that enrollment events have no associated `reason`.

`results` — Optional — Yes. Multiple values create an `OR` query.

The result of an authentication attempt. One of:

Value	Description
`success`	Return "successful" authentication events.
`denied`	Return "denied" authentication events.
`fraud`	Return "fraudulent" authentication events.

Default: Return logs for any result. Filtering on all values is equivalent to the default.

`tokens` — Optional — Yes. Multiple values create an `OR` query.

A WebAuthn security key's `webauthnkey` or U2F security key's `registration_id` as returned in the authentication log output.

Default: Return logs for security keys used.

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid or missing parameters or `mintime` is after the `maxtime`.

RESPONSE FORMAT

Key	Value

`access_device`

Browser, plugin, and operating system information for the endpoint used to access the Duo-protected resource. Values present only when the application accessed features Duo's inline browser prompt.

This information is available to Duo Premier and Duo Advantage plan customers.

`browser`	The web browser used for access.
`browser_version`	The browser version.
`epkey`	The endpoint's unique identifier. Most reliable when reported by Duo Desktop installed on the endpoint.
`flash_version`	The Flash plugin version used, if present, otherwise "uninstalled".
`hostname`	The hostname, if present, otherwise `null` .
`ip`	The access device's IP address, if present, otherwise `null` .
`is_encryption_enabled`	Reports the disk encryption state as detected by Duo Desktop. One of `true` , `false` , or "unknown".
`is_firewall_enabled`	Reports the firewall state as detected by Duo Desktop. One of `true` , `false` , or "unknown".
`is_password_set`	Reports the system password state as detected by Duo Desktop. One of `true` , `false` , or "unknown".
`java_version`	The Java plugin version used, if present, otherwise "uninstalled".
`location`	The GeoIP location of the access device, if available. The response may not include all location parameters. `city` — The city name. `country` — The country name. Refer to ISO 3166 for a list of possible countries. `state` — The state, county, province, or prefecture.
`os`	The device operating system name.
`os_version`	The device operating system version.
`security_agents`	Reports the security agents present on the endpoint as detected by Duo Desktop.

`adaptive_trust_assessments`

Risk-based authentication information. Values present only when the application accessed features Duo's inline browser prompt and has a Duo Risk-Based Authentication policy applied.

This information is available to Duo Premier and Duo Advantage plan customers.

Type of adaptive trust assessment. One of:

`more_secure_auth` : Trust assessment information for Risk-Based Factor Selection.

`remember_me` : Trust assessment information for Risk-Based Remembered Devices.

`detected_attack_detectors`	List of the risk-based authentication detections found during or after an authentication attempt. Only returned for `more_secure_auth` .
`features_version`	The feature version for the risk-based authentication trust assessment.

`model_version`	The model version for the risk-based authentication trust assessment.
`policy_enabled`	Denotes if risk-based authentication was enabled by the policy under which the trust assessment was evaluated. One of: `true`, `false`, or `false` (reserved for historical authentication logs that do not have the `policy_enabled` field populated).
`reason`	The reason behind the trust assessment level.
`trust_level`	The trust assessment level. One of: `ERROR`, `LOW`, `NORMAL`, `UNKNOWN`, or `UNSET`.

`alias`	The username alias used to log in. No value if the user logged in with their `username` instead of a username alias.
`application`	Information about the application accessed. `key` — The application's `integration_key`. `name` — The application's name.
`auth_device`	Information about the device used to approve or deny authentication. `ip` — The IP address of the authentication device. `key` — The Duo identifier of the authentication device (the `phone_id` value for a phone, the `webauthnkey` value for a security key, etc.). `location` — The GeoIP location of the authentication device, if available. The response may not include all location parameters. `city` — The city name. `country` — The country name. Refer to ISO 3166 for a list of possible countries. `state` — The state, county, province, or prefecture. `name` — The name of the authentication device.
`email`	The email address of the user, if known to Duo, otherwise none.
`event_type`	The type of activity logged. one of: "authentication" or "enrollment".
`factor`	The authentication factor. One of: "bypass_code", "Desktop Authenticator", "digipass_go_7_token", "duo_mobile_passcode", "duo_mobile_passcode_hotp", "duo_mobile_passcode_totp", "duo_push", "hardware_token", "not_available", "passcode", "phone_call", "Platform authenticator (2fa)", "remembered_device", "Roaming authenticator (2fa)", "sms_passcode", "sms_refresh", "trusted_mobile_authenticator", "trusted_network" "u2f_token", "verified_duo_push", "WebAuthn Chrome Touch ID", "WebAuthn Credential", "WebAuthn Security Key", or "yubikey_passcode".
`isotimestamp`	ISO8601 timestamp of the event.

`ood_software`	If authentication was denied due to out-of-date software, shows the name of the software, i.e. "Chrome", "Flash", etc. No value if authentication was successful or authentication denial was not due to out-of-date software.
`passport_assessment`	Information on whether the authentication supported Duo Passport. Will return for all authentications, even if Duo Passport is not enabled. Authentication logs before August 2024 will not return this field. `is_supported`: If `true`, the authentication supported Duo Passport. If `false`, the authentication did not support Duo Passport. `reason`: Returns `supported` if `is_supported` is `true`. Otherwise, returns a reason on why Duo Passport could not be used for that authentication.
`reason`	Provide the reason for the authentication attempt result. If result is "SUCCESS" then one of: "allow_unenrolled_user", "allowed_by_policy", "allow_unenrolled_user_on_trusted_network", "bypass_user", "remembered_device", "trusted_location", "trusted_network", "user_approved", "valid_passcode". If result is "FAILURE" then one of: "anonymous_ip", "could_not_determine_if_endpoint_was_trusted", "denied_by_policy", "denied_network", "deny_unenrolled_user", "endpoint_is_not_in_management_system", "endpoint_failed_google_verification", "endpoint_is_not_trusted", "factor_restricted", "frequent_attempts", "invalid_management_certificate_collection_state", "invalid_device", "invalid_passcode", "invalid_referring_hostname_provided", "location_restricted", "locked_out", "no_activated_duo_mobile_account", "no_disk_encryption", "no_duo_certificate_present", "touchid_disabled", "no_referring_hostname_provided", "no_response", "no_screen_lock", "no_web_referer_match", "out_of_date", "platform_restricted", "rooted_device", "software_restricted", "user_cancelled", "user_disabled", "user_mistake", "user_not_in_permitted_group", "user_provided_invalid_certificate", or "version_restricted". If result is "ERROR" then: "error". If result is "FRAUD" then: "user_marked_fraud". Note that enrollment events have no associated `reason`.
`result`	The result of the authentication attempt. One of: "success", "denied", "failure", "error", or "fraud".
`timestamp`	An integer indicating the Unix timestamp of the event.
`txid`	The transaction ID of the event.
`user`	Information about the authenticating user. `groups`: Duo group membership information for the user. `key`: The user's `user_id`. `name`: The user's `username`.

EXAMPLE RESPONSE

```
200 OK
{
    "stat": "OK",
    "response": {
        "authlogs": [
```

```
{
    "access_device": {
        "browser": "Chrome",
        "browser_version": "67.0.3396.99",
        "epkey": "EP18JX1A10AB102M2T2X",
        "flash_version": "uninstalled",
        "hostname": null,
        "ip": "169.232.89.219",
        "is_encryption_enabled": true,
        "is_firewall_enabled": true,
        "is_password_set": true,
        "java_version": "uninstalled",
        "location": {
            "city": "Ann Arbor",
            "country": "United States",
            "state": "Michigan"
        },
        "os": "Mac OS X",
        "os_version": "10.14.1",
        "security_agents": []
    },
    "adaptive_trust_assessments": {
        "more_secure_auth": {
            "detected_attack_detectors": [
                "USER_MARKED_FRAUD"
            ],
            "features_version": "3.0",
            "model_version": "2022.07.19.001",
            "policy_enabled": true,
            "reason": "Low level of trust; detection of one or more known attack patte
            "trust_level": "LOW"
        },
        "remember_me": {
            "features_version": "3.0",
            "model_version": "2022.07.19.001",
            "policy_enabled": false,
            "reason": "Known Access IP",
            "trust_level": "NORMAL"
        }
    },
    "alias": "",
    "application": {
        "key": "DIY231J8BR23QK4UKBY8",
        "name": "Microsoft Azure Active Directory"
    },
    "auth_device": {
        "ip": "192.168.225.254",
        "key": "DP5BJ05HI4WRBVI4Q7JF",
        "location": {
            "city": "Ann Arbor",
            "country": "United States",
            "state": "Michigan"
        },
        "name": "My iPhone X (734-555-2342)"
    },
    "email": "narroway@example.com",
    "event_type": "authentication",
    "factor": "duo_push",
    "isotimestamp": "2020-02-13T18:56:20.351346+00:00",
    "ood_software": null,
    "passport_assessment": {
        "is_supported": false,
```

```
                "reason": "absent_health_report"
            },
            "reason": "user_approved",
            "result": "success",
            "timestamp": 1581620180,
            "trusted_endpoint_status": "not trusted",
            "txid": "340a23e3-23f3-23c1-87dc-1491a23dfdbb",
            "user": {
                "groups": [
                    "Duo Users",
                    "CorpHQ Users"
                ],
                "key": "DU3KC77WJ06Y5HIV7XKQ",
                "name": "narroway@example.com"
            }
        },
    ],
    "metadata": {
        "next_offset": [
            "1532951895000",
            "af0ba235-0b33-23c8-bc23-a31aa0231de8"
        ],
        "total_objects": 1
    }
},
}
```


Authentication Logs (Legacy v1)

The v2 authentication logs endpoint provides new filtering and querying capabilities unavailable in the legacy v1 handler, like the ability to filter on users, groups, applications, authentication results, factors, and time ranges. Consider migrating to the new handler.

Returns a list of authentication log events ranging from the last 180 days up to as recently as two minutes before the API request. There is an intentional two minute delay in availability of new authentications in the API response. Duo operates a large scale distributed system, and this two minute buffer period ensures that calls will return consistent results. Querying for results more recent than two minutes will return as empty. Requires "Grant read log" API permission.

The 1000 earliest events will be returned; you may need to call this multiple times with `mintime` to page through the entire log. Note that more or fewer than 1000 events may be returned depending on how many actual events exist for the specified `mintime`.

We recommend requesting logs no more than once per minute.

`GET` `/admin/v1/logs/authentication`

PARAMETERS

Parameter	Required?	Description
`mintime`	Optional	Only return records that have a Unix `timestamp` in seconds of `mintime` or later. Use `mintime+1` to avoid receiving duplicate data.

RESPONSE CODES

Response	Meaning
200	Success.

RESPONSE FORMAT

Key	Value
`access_device`	Browser, plugin, and operating system information for the endpoint used to access the Duo-protected resource. Values present only when the application accessed features Duo's inline browser prompt. This information is available to Duo Premier and Duo Advantage plan customers. `browser` — The web browser used for access. `browser_version` — The browser version. `flash_version` — The Flash plugin version used, if present, otherwise "uninstalled". `java_version` — The Java plugin version used, if present, otherwise "uninstalled". `os` — The device operating system name. `os_version` — The device operating system version. `trusted_endpoint_status` — Shows whether the endpoint is a Duo managed endpoint. One of "trusted", "not trusted", "unknown", or "error". Returned for Duo Premier customers only.
`alias`	The username alias used to log in. No value if the user logged in with their `username` instead of a username alias.
`device`	Device used to authenticate, if present, otherwise none.
`email`	The email address of the user, if known to Duo, otherwise none.
`factor`	The authentication factor. One of: "Bypass Code", "Digipass GO 7 Token", "Duo Desktop", "Duo Mobile Inline Auth", "Duo Mobile Passcode", "Duo Mobile Passcode (HOTP)", "Duo Mobile Passcode (TOTP)", "Duo Push", "Passcode", "Phone Call", "Hardware Token", "Remembered Device", "Security Key (WebAuthn)", "SMS Passcode", "SMS Refresh", "Touch ID (WebAuthn)", "Trusted Network", "U2F Token", or "Yubikey Passcode".
`integration`	The integration name.
`ip`	The IP address that this request originated from.
`isotimestamp`	ISO8601 timestamp of the event.
`location`	The GeoIP location from which the user authenticated, if available. The response may not include all location parameters. `city` — The city name. `state` — The state, county, province, or prefecture. `country` — The country code. Refer to ISO 3166 for a list of possible countries.
`new_enrollment`	`true` if the user enrolled a new device at the authentication prompt; `false` if authenticated with a previously enrolled device.
`ood_software`	If authentication was denied due to out-of-date software, shows the name of the software, i.e. "Chrome", "Flash", etc. No value if authentication was successful or authentication denial was not due to out-of-date software.
`reason`	Provide the reason for the authentication attempt result. If result is "SUCCESS" then one of: "Allow unenrolled user", "Allowed by policy", "Bypass user", "Remembered device", "Trusted location", "Trusted network", "User approved", "Valid passcode". If result is "FAILURE" then one of: "Anonymous IP", "Call timed out", "Couldn't determine if endpoint was trusted", "Denied by policy", "Deny unenrolled user", "Duo Mobile version restricted", "Endpoint is not in Management System", "Factor restricted", "Frequent attempts", "Invalid device", "Invalid

	passcode", "Location restricted", "Locked out", "No Duo certificate present", "No response", "No disk encryption", "No fingerprint", "No screen lock", "Out of date", "Platform restricted", "Rooted device", "Software Restricted", "User cancelled", "User is disabled", "User mistake", or "User provided an invalid certificate". If result is "ERROR" then: "Error". If result is "FRAUD" then: "User marked fraud". Note that enrollment events have no associated `reason`.
`result`	The result of the authentication attempt. One of: "success", "denied", "failure", "error", or "fraud".
`timestamp`	An integer indicating the Unix timestamp of the event.
`username`	The authenticating user's username.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": [{
      "access_device": {
          "browser": "Chrome",
          "browser_version": "56.0.2924.87",
          "flash_version": "25.0.0.0",
          "java_version": "1.8.0.92",
          "os": "Mac OS X",
          "os_version": "10.11"
          "trusted_endpoint_status": "trusted"
      },
      "alias": "john.smith",
      "device": "503-555-1000",
      "email": "jsmith@example.com",
      "factor": "Duo Push",
      "integration": "Acme Intranet",
      "ip": "192.168.0.1",
      "isotimestamp": "2020-02-13T18:56:20.351346+00:00",
      "location": {
          'city': 'Ann Arbor',
          'state': 'Michigan',
          'country': 'US'
      },
      "new_enrollment": false,
      "ood_software": "",
      "reason": "User approved",
      "result": "SUCCESS",
      "timestamp": 1581620180,
      "username": "jsmith",

  }]
}
```


Activity Logs

This API endpoint is currently in Public Preview.

Returns a paged list of activity log events ranging from the last 180 days up to as recently as two minutes before the API request. The events returned are subject to log retention, if set up in the account, as described here. To fetch all results, call repeatedly with the `next_offset` paging parameter as long as the result metadata has `next_offset` values. Requires "Grant read log" API permission.

There is an intentional two-minute delay in the availability of new activity events in the API response. Duo operates a large-scale distributed system, and this two-minute buffer period ensures that calls will return consistent results. Querying for results more recent than two minutes will return as empty.

We recommend requesting logs no more than once per minute.

`GET` `/admin/v2/logs/activity`

PARAMETERS

Query Parameter	Required?	Allow Multiple?	Description
`mintime`	**Required**	No	Return records that have a 13 character Unix `timestamp` in milliseconds of `mintime` or later. This value must be strictly less than `maxtime`. Example: `1661022959934`
`maxtime`	**Required**	No	Return records that have a 13 character Unix `timestamp` in milliseconds of `maxtime` or earlier. This value must be strictly greater than `mintime`. Maximum: 180 days Example: `1661022969934`
`limit`	Optional	No	The maximum number of records returned. Default: `100`; Max: `1000`
`next_offset`	Optional	Yes	The offset at which to start record retrieval. This value is provided in the metadata in the form of a 13 character date string in milliseconds and the event `txid`. Both of these values must be provided when used, separated by a comma. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: None Example: `next_offset=1547486297000,5bea1c1e-612c-4f1d-b310-75fd31385b15`
`sort`	Optional	No	The order in which to return records. One of: **Value** / **Description** `ts:asc` — Return logs in chronological order. `ts:desc` — Return logs in reverse chronological order. Default: `ts:desc`

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid or missing parameters or `mintime` is after the `maxtime`.

RESPONSE FORMAT

Key	Value
`items`	List of activity logs. Information included:

`access_device`

Information about the device used to perform the activity, to the extent that we can capture it. A are optional as non-external actors will not have any information to record here.

`browser`	The web browser used for access.
`browser_version`	The web browser version.
`epkey`	The device's unique identifier or epkey.
`ip`	IP information of the access device. `address` — IP address of access device.
`location`	The location details of the device, if available. `city` — The city name. `country` — The country code. Refer to ISO 3166 for a list of possible countries. `state` — The state, county, province, or prefecture.
`os`	The device operating system name.
`os_version`	The device operating system version.

`action`

Information about the action performed.

`details`	Provides additional information about the action. Details is optional.
`name`	The `name` is a string representing the action the `actor` performed. If a target present, the action was performed on that target. One of:

Action	Desc
`admin_activate_duo_push`	An adminis user ac 2FA de Duo Pu
`admin_factor_restrictions`	An adminis update restricti
`admin_login`	Adminis logged
`admin_reactivates_duo_push`	Adminis reactiva admin c 2FA de Duo Pu
`admin_reset_password`	Adminis change own pa
`admin_send_reset_password_email`	Passwo email s adminis
`azure_directory_create`	Entra ID director created
`azure_directory_delete`	Entra ID director deleted
`azure_directory_update`	Entra ID director update

`azure_sync_begin`	Entra ID sync be
`azure_sync_fail`	Entra ID sync fa
`azure_sync_finish`	Entra ID sync fin
`bypass_create`	An adminis created more by codes.
`bypass_delete`	An adminis deleted bypass
`bypass_view`	An adminis viewed bypass
`cloudsso_add_authproxy`	Added Authen Proxy fo Sign-On
`cloudsso_add_bridge_attribute`	Added attribute Single S
`cloudsso_add_email_domain`	Added Sign-On Domain
`cloudsso_add_ldap_authsource`	Added A Directo authent source Single S
`cloudsso_add_ldap_authsource_and_accept_tcs`	Added A Directo authent source Single S and acc terms & conditio
`cloudsso_add_saml_authsource`	Added authent source Single S
`cloudsso_create_launcher`	Created Central
`cloudsso_create_routing_rule`	Create rule for Sign-On
`cloudsso_create_tiles`	Added Central
`cloudsso_delete_authproxy`	Deleted Authen Proxy fo Sign-On
`cloudsso_delete_bridge_attribute`	Deleted attribute Single S
`cloudsso_delete_email_domain`	Deleted Sign-On Domain
`cloudsso_delete_ldap_authsource`	Deleted Directo authent source Single S

`cloudsso_delete_routing_rule`	Delete rule for Sign-O
`cloudsso_delete_saml_authsource`	Deletec authent source Single S
`cloudsso_delete_tile`	Deletec Central
`cloudsso_disable_authsource`	Disable authent source Single S
`cloudsso_enable_authsource`	Enable authent source Single S
`cloudsso_enable_single_sign_on`	Enable Sign-O
`cloudsso_modify_tile`	Modifie Central
`cloudsso_oauth_cc_get_client_secret`	Viewed 2.0 clie credent client s
`cloudsso_oauth_cc_reset_client_secret`	Reset C 2.0 clie credent client s
`cloudsso_oidc_get_client_secret`	Viewed relying client s
`cloudsso_oidc_reset_client_secret`	Reset C relying client s
`cloudsso_update_bridge_attribute`	Update attribut Single S
`cloudsso_update_launcher`	Update Central
`cloudsso_update_ldap_authsource`	Update Sign-O Directo configu
`cloudsso_update_oauth_client_credentials`	Update generic 2.0 SS integrat
`cloudsso_update_relying_party`	Update generic SSO integrat
`cloudsso_update_routing_rule`	Update rule for Sign-O
`cloudsso_update_saml_authsource`	Update authent source Single S
`cloudsso_verify_email_domain`	Verified Sign-O Domain
`deregister_devices`	De-regi devices includin associa users, t

	registe Duo De
`device_change_enrollment_summary_notification_answered`	Device enrollm summa notifica answer user.
`device_change_enrollment_summary_` `notification_answered_notify_admin`	Device enrollm summa notifica answer user an sent to
`device_change_enrollment_summary_notification_send`	Device enrollm summa notifica sent.
`device_change_notification_answered`	Device notifica answer user.
`device_change_notification_answered_notify_admin`	Device notifica answer user an sent to
`device_change_notification_create`	Device notifica created
`device_change_notification_send`	Device notifica sent.
`group_create`	An adminis created group.
`group_delete`	An adminis deleted group.
`group_update`	An adminis updated group.
`hardtoken_create`	Created hardwa token.
`hardtoken_delete`	Deletec hardwa token.
`hardtoken_resync`	Resync a hardw token.
`hardtoken_update`	Modifie hardwa token.
`integration_create`	An adminis created applica
`integration_delete`	An adminis deleted applica
`integration_group_policy_add`	An adminis group p

	was ad the app
`integration_group_policy_remove`	An adminis group p was rer from th applica
`integration_policy_assign`	An adminis assigne policy t applica
`integration_policy_unassign`	An adminis unassig policy fi applica
`integration_skey_bulk_view`	Secret I present plain te
`integration_skey_view`	Applica secret K copied text in t Admin P viewed Admin A plain te
`integration_update`	An adminis updated applica
`log_export_start`	Log exp started.
`log_export_complete`	Log exp comple
`log_export_failure`	Log exp failed.
`management_system_activate_device_cache`	Duo Tru Endpoi device activate
`management_system_active_device_cache_add_devices`	Devices to an ac Duo Tru Endpoi device
`management_system_active_device_cache_delete_devices`	Devices from an Duo Tru Endpoi device
`management_system_active_device_cache_edit_devices`	Devices in an ac Duo Tru Endpoi device
`management_system_add_devices`	Devices to a Du Trusted Endpoi device
`management_system_create`	Duo Tru Endpoi integrat created
`management_system_delete`	Duo Tru Endpoi integrat deleted

`management_system_delete_devices`	Devices from a l Trusted Endpoi device
`management_system_device_cache_add_devices`	Devices to a Du Trusted Endpoi device
`management_system_device_cache_create`	Duo Tru Endpoi device created
`management_system_device_cache_delete`	Duo Tru Endpoi device deleted
`management_system_device_cache_delete_devices`	Devices from Du Trusted Endpoi device
`management_system_download_device_api_script`	Duo De script downlo
`management_system_pkcs12_enrollment`	Duo Tru Endpoi certifica enrolle
`management_system_sync_failure`	Duo Tru Endpoi integrat failed.
`management_system_sync_success`	Duo Tru Endpoi integrat succee
`management_system_update`	Duo Tru Endpoi integrat update
`management_system_view_password`	Duo Tru Endpoi integrat passwo viewed.
`management_system_view_token`	Duo Tru Endpoi integrat token v
`phone_activation_code_regenerated`	An activ code ha generat phone.
`phone_associate`	Existing has bee added t user.
`phone_create`	Added
`phone_delete`	A phon users associa it has b deleted

`phone_disassociate`	An exis phone been re from a still has users a
`phone_new_sms_passcode`	SMS co been g and ser user.
`phone_update`	Modifie
`policy_create`	Create policy.
`policy_delete`	Delete policy.
`policy_update`	Modifie policy.
`u2ftoken_create`	Create token.
`u2ftoken_delete`	Delete token.
`user_adminapi_lockout`	User wa locked Admin A
`user_create`	User cr
`user_delete`	User de
`user_import`	User im
`user_lockout_cleared`	User ha locked status c
`user_not_enrolled_lockout`	User wa unenro beyond number specifie "Unenr users" s
`user_pending_delete`	Delete pendin
`user_restore`	Delete restore
`user_update`	Existing details
`webauthncredential_create`	Create WebAu authent and ass it with t
`webauthncredential_delete`	Delete WebAu authent
`webauthncredential_rename`	Renam WebAu authent

`activity_id`	Transaction ID of the event.
`actor`	The `actor` represents the entity performing the action.

`details`	If actor `type` is `user` or `admin`, following fields would be populated:

`created`	Timestamp when actor was created.
`group`	The group membership of the actor, if actor `type` is `user`. `key` — The group's integration key. `name` — The group's name.
`last_login`	Timestamp of last login.
`status`	Status of actor. One of: `Active`, `Bypass`, `Disabled`, or `Locked Out`.

`key`	Identifier of the actor.
`name`	Name of the actor.
`type`	Type of actor. One of: `admin`, `adminapi`, `admin_sync`, `azure_sync`, `dev`, `ldapsync`, `system`, or `user`.

`akey`	Unique identifier of entity associated with the activity log.
`application`	The integration used to perform the activity. `key` — The application's integration key. `name` — The application's name. `type` — The application's type.
`old_target`	The `target` represents the entity on which the `action` was performed, before the `action` performed. The `old_target` is optional, as not all activity may represent a change to another `details` — Key-value pair of properties about the target. The properties for a given target may by target type, but should be consistent for the same type. `key` — Key of the target that corresponds to the target type. `name` — Name of the target. `type` — The target type. One of: `admin`, `adminap_integrations`, `authproxy`, `computer_registration`, `device_registration`, `enroll_code`, `group`, `log_export`, `login_settings`, `hardtoken`, `integration`, `phone`, `po`, `trusted_endpoints_integration`, `u2f_token`, `user`, `user_bypass`, or `webauthn_credentials`.
`outcome`	Result of the `ADMIN_ACTION_ADMIN_LOGIN` action. By default, the `outcome` field is `"SUCC` failure, the `outcome` field is `"FAILURE"`.
`target`	The `target` represents the entity on which the `action` was performed. The target is optional, all activity may represent a change to another entity. `details` — Key-value pair of properties about the target. The properties for a given target may by target type, but should be consistent for the same type. `key` — Key of the target that corresponds to the target type. `name` — Name of the target. `type` — The target type. One of: `admin`, `adminap_integrations`, `authproxy`, `computer_registration`, `device_registration`, `enroll_code`, `group`, `log_export`, `login_settings`, `hardtoken`, `integration`, `phone`, `po`, `trusted_endpoints_integration`, `u2f_token`, `user`, `user_bypass`, or `webauthn_credentials`.

`ts`	The time at which the activity took place, to the best of our ability to record it. This may not alw the exact time the event took place, but should be extremely close.

EXAMPLE RESPONSE


```
{
  "response": {
        "items": [
            {
                "access_device": {
                    "browser": "Chrome",
                    "browser_version": "111.0.0.0",
                    "epkey": "EP123456789012345678",
                    "ip": {
                        "address": "172.34.40.116"
                    },
                    "location": {
                        "city": "Ann Arbor",
                        "country": "United States",
                        "state": "Michigan"
                    },
                    "os": "Mac OS X",
                    "os_version": "10.15.7"
                },
                "action": {
                    "details": null,
                    "name": "webauthncredential_create"
                },
                "activity_id": "720b8360-078b-47c4-adc7-7968df1caef0",
                "actor": {
                    "details": "{\"created\": \"2015-09-25T23:17:40.000000+00:00\", \"last_login\
                    \"2023-03-21T19:51:09.000000+00:00\", \"status\": \"Active\", \"groups\": [{\
                    \"CorpHQ_Users\", \"key\": \"DGAZ172QBWDM26AK8ITK\"}, {\"name\": \"ITAdmins\"
                    \"DGK3B7XTSIP00LKHK1RD\"}, {\"name\": \"yee\", \"key\": \"DGKZWSBCDADEVFGFK5N
                    "key": "DU64TKJPJ0SHFWKO2LNBC",
                    "name": "sogilby",
                    "type": "user"
                },
                "akey": "DAAR5FO0OZ4VYZA0WOB2",
                "application": {
                    "key": "DILSVDEYH66TBHKIXGR9",
                    "name": "Acme Corp",
                    "type": "websdk"
                },
                "old_target": null,
                "outcome": {
                    "result"; "FAILURE"
                }
                "target": {
                    "details": "{\"authenticator_type\": \"Security key\", \"transport_types\": \
                    \"passwordless_authorized\": false, \"browser\": \"Chrome\", \"browser_versio
                    \"111.0.0.0\", \"os\": \"Mac OS X\", \"os_version\": \"10.15.7\", \"user_agen
                    \"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML,
                    Chrome/111.0.0.0 Safari/537.36\", \"credential_name\": \"Security key\"}",
                    "key": "WAUKH0IMTGP00L90LT4KM",
                    "name": "WAUKH0IMTG3EDD4DT4KM",
                    "type": "webauthn_credential"
                },
                "ts": "2023-03-21T15:51:22.591015+00:00"
            }
```

```
        ],
        "metadata": {
            "next_offset": "1666714065304,5bf1a860-fe39-49e3-be29-217659663a74",
            "total_objects": 1
        }
    },
  "stat": "OK",
}
```


Administrator Logs

Returns a list of administrator log events. Only the 1000 earliest events will be returned; you may need to call this multiple times with `mintime` to page through the entire log. Requires "Grant read log" API permission.

We recommend requesting logs no more than once per minute.

`GET` `/admin/v1/logs/administrator`

PARAMETERS

Parameter	Required?	Description
`mintime`	Optional	Only return records that have a Unix `timestamp` in seconds after `mintime`. This can help to avoid fetching records that have already been retrieved.

RESPONSE CODES

Response	Meaning
200	Success.

RESPONSE FORMAT

Key	Value
`action`	The type of change that was performed. Possible values: **Type** / **Description** "accepted_dtm_opt_in" — Admin accepted Duo Trust Monitor terms "accountsapi_request_ip_denied" — Accounts API request blocked due to IP restriction "activation_begin" — New admin begins setup using activation link "activation_complete" — New admin completes account setup "activation_create_link" — New admin activation link created "activation_delete_link" — Admin activation link deleted "activation_send_email" — Emailed activation link to admin "activation_set_password" — New admin sets login password during initial setup "ad_connection_config_download" — Admin downloaded authproxy.cfg file for an Active Directory sync connection "ad_sync_begin" — Active Directory Sync started "ad_sync_by_user_begin" — Active Directory Sync of individual user started "ad_sync_by_user_finish" — Active Directory Sync of individual user completed "ad_sync_config_download" — Active Directory Sync configuration download "ad_sync_failed" — Active Directory Sync failed

"ad_sync_finish"	Active Directory Sync completed
"admin_2fa_error"	Administrator issue completing secondary authentication
"admin_account_switch"	Administrator switched to this account from another Duo account
"admin_activate_duo_push"	Administrator activates an admin's 2FA device for Duo Push
"admin_activation_create"	An admin activation link was created
"admin_activation_delete"	An admin activation link was deleted
"admin_create"	Added administrator
"admin_delete"	Deleted administrator
"admin_lockout"	Administrator locked out after too many failed login attempts
"admin_login_error"	Administrator issue completing primary password or SAML authentication
"admin_login"	Administrator logged in
"admin_purchase_hardtokens"	Administrator purchased hardtokens
"admin_reactivates_duo_push"	Administrator reactivates an admin's 2FA device for Duo Push
"admin_reset_password"	Administrator changed password via emailed reset link
"admin_restore"	Administrator controlled by an admin directory sync is restored from pending deletion
"admin_scso_enforcement_change"	Updated Cisco Security Cloud Sign On (SCSO) enforcement status
"admin_self_activate"	Administrator self-activated through an activation link
"admin_send_reset_password_email"	Password reset email sent to administrator
"admin_single_sign_on_cert_create"	Generated a new certificate for admin SSO
"admin_single_sign_on_cert_delete"	Deleted a certificate for admin SSO
"admin_single_sign_on_cert_update"	Modified a certificate for admin SSO
"admin_single_sign_on_update"	Modified admin SSO configuration
"admin_sync_create"	Created an admin directory sync
"admin_sync_delete"	Deleted an admin directory sync
"admin_sync_pause"	Paused an admin directory sync
"admin_sync_resume"	Resumed an admin directory sync
"admin_sync_update"	Updated an admin directory sync
"admin_unexpire"	Expired administrator's inactivity was cleared
"admin_update"	Modified administrator
"admin_view_password_reset_url"	Viewed the "Forgot Password" URL
"adminapi_request_ip_denied"	Admin API request blocked due to IP restriction
"administrative_unit_create"	Added an administrative unit
"administrative_unit_delete"	Deleted an administrative unit
"administrative_unit_update"	Updated an administrative unit
"authproxy_hide"	An Authentication Proxy was hidden on the Authentication Proxy Dashboard

"authproxy_show"	An Authentication Proxy was unhidden and shown on the Authentication Proxy Dashboard
"authproxy_update"	An Authentication Proxy's name and/or description was updated on the Authentication Proxy Dashboard
"blacklist_trusted_endpoint"	Endpoint added to deny list
"branding_draft_delete"	Deleted draft custom branding
"branding_draft_update"	Updated draft custom branding
"branding_draft_users_update"	Updated test users for draft custom branding
"branding_update"	Updated custom branding
"bypass_create"	Bypass code created for a user
"bypass_delete"	Bypass code deleted
"bypass_view"	Viewed a user's bypass code
"cloudsso_add_authproxy"	Added an Authentication Proxy for Single Sign-On
"cloudsso_add_bridge_attribute"	Added a bridge attribute for Single Sign-On
"cloudsso_add_email_domain"	Added a Single Sign-On email domain
"cloudsso_add_ldap_authsource"	Added Active Directory authentication source for Single Sign-On
"cloudsso_add_saml_authsource"	Added SAML authentication source for Single Sign-On
"cloudsso_create_launcher"	Created Duo Central
"cloudsso_create_tiles"	Added Duo Central tile(s)
"cloudsso_delete_authproxy"	Deleted an Authentication Proxy for Single Sign-On
"cloudsso_delete_bridge_attribute"	Deleted a bridge attribute for Single Sign-On
"cloudsso_email_domain"	Deleted a Single Sign-On email domain
"cloudsso_delete_ldap_authsource"	Deleted Active Directory authentication source for Single Sign-On
"cloudsso_delete_saml_authsource"	Deleted SAML authentication source for Single Sign-On
"cloudsso_delete_tile"	Deleted Duo Central tile
"cloudsso_disable_authsource"	Disabled authentication source for Single Sign-On
"cloudsso_enable_authsource"	Enabled authentication source for Single Sign-On
"cloudsso_enable_single_sign_on"	Enabled Single Sign-On
"cloudsso_modify_tile"	Modified Duo Central tile
"cloudsso_oauth_cc_get_client_secret"	Viewed OAuth 2.0 client credentials client secret
"cloudsso_oauth_cc_reset_client_secret"	Reset OAuth 2.0 client credentials client secret
"cloudsso_oidc_get_client_secret"	Viewed OIDC relying party client secret
"cloudsso_oidc_reset_client_secret"	Reset OIDC relying party client secret
"cloudsso_update_bridge_attribute"	Updated a bridge attribute for Single Sign-On
"cloudsso_update_launcher"	Updated Duo Central
"cloudsso_update_ldap_authsource"	Updated Single Sign-On Active Directory configuration

"cloudsso_update_oauth_client_credentials"	Updated a generic OAuth 2.0 SSO integration
"cloudsso_update_relying_party"	Updated a generic OIDC SSO integration
"cloudsso_update_saml_authsource"	Updated Single Sign-On SAML configuration
"cloudsso_update_service_provider"	Updated "Service Provider" information for an SSO integration
"cloudsso_verify_email_domain"	Verified a Single Sign-On email domain
"create_child_customer"	Created child customer
"credits_update"	Added telephony credits
"custom_messaging_update"	Updated custom help desk message in the global settings
"customer_update"	Modified settings
"delete_child_customer"	Deleted child customer
"delete_enroll_code"	Deleted an enrollment link
"deregister_devices"	Deregistered device(s)
"deviceapi_request_ip_denied"	Device API request blocked due to IP restriction
"directory_create"	Added directory for sync
"directory_delete"	Deleted directory
"directory_groups_update"	Modified directory sync groups
"directory_sync_pause"	Scheduled directory sync paused by administrator or due to expired Entra ID authorization
"directory_sync_resume"	Scheduled directory sync resumed by administrator or after Entra ID reauthorization
"directory_update"	Modified directory sync
"dtm_modify_user"	Modified user from Trust Monitor
"edition_update"	Update edition
"entra_dir_connection_authorize"	Authorized Entra ID directory connection
"entra_directory_create"	Added Entra ID directory
"entra_directory_delete"	Deleted Entra ID directory
"entra_directory_groups_update"	Modified Entra ID directory groups
"entra_directory_update"	Modified Entra ID directory
"entra_sync_begin"	Full Entra ID Sync started
"entra_sync_by_user_begin"	Entra ID Sync started (sync by username)
"entra_sync_by_user_finish"	Entra ID Sync completed (sync by username)
"entra_sync_finish"	Entra ID Sync completed
"entra_sync_fail"	Entra ID Sync failed
"feature_add"	Added feature
"feature_delete"	Deleted feature
"group_create"	Added group
"group_delete"	Deleted group
"group_update"	Updated group
"hardtoken_create"	Created hardware tokens
"hardtoken_delete"	Deleted hardware token

"hardtoken_resync"	Resynchronized hardware token
"hardtoken_update"	Modified hardware token
"inactive_admin_expiration_change"	Changed inactive admins setting
"integration_create"	Added application
"integration_delete"	Deleted application
"integration_download_saml_verification_public_key"	Downloaded public key for SAML verification
"integration_group_policy_add"	Add application group policy
"integration_group_policy_remove"	Remove application group policy
"integration_group_policy_reorder"	Reordered application group policies
"integration_group_policy_update"	Modified application group policy members
"integration_policy_assign"	Assign application policy
"integration_policy_unassign"	Unassign application policy
"integration_skey_bulk_view"	Viewed application's secret key or client ID via Admin API
"integration_skey_view"	Viewed application's secret key or client ID in the Admin Panel
"integration_sso_saml_metadata_import"	Imported SAML metadata for an SSO integration
"integration_universal_prompt_update"	Update integration to use the Universal Prompt from the Universal Prompt Progress report page
"integration_update"	Modified application
"ldap_connection_skey_reset"	Reset secret key for Active Directory or OpenLDAP sync connection
"ldap_connection_skey_view"	View secret key for Active Directory or OpenLDAP sync connection
"ldap_directory_conn_create"	Created an OpenLDAP sync connection
"ldap_directory_conn_delete"	Deleted an OpenLDAP sync connection
"ldap_directory_conn_modify"	Modified an OpenLDAP sync connection
"management_system_activate_device_cache"	Duo Trusted Endpoints device cache activated
"management_system_add_devices"	Devices added to a Duo Trusted Endpoints device cache
"management_system_create"	Duo Trusted Endpoints integration created
"management_system_delete_device_cache"	Duo Trusted Endpoints device cache deleted
"management_system_delete_devices"	Devices deleted from a Duo Trusted Endpoints device cache
"management_system_delete"	Duo Trusted Endpoints integration deleted
"management_system_download_device_api_script"	Duo Device API script downloaded
"management_system_sync_failure"	Duo Trusted Endpoints integration sync failed
"management_system_sync_success"	Duo Trusted Endpoints integration sync succeeded
"management_system_update"	Duo Trusted Endpoints integration updated
"management_system_view_password"	Duo Trusted Endpoints integration password viewed
"management_system_view_token"	Duo Trusted Endpoints integration token viewed

"migrate_dag_app"	Converted Duo Access Gateway application to Duo Single Sign-On
"oort_accepted_pds"	Accepted Cisco Identity Intelligence privacy statement
"oort_credentials_create"	Cisco Identity Intelligence OIDC credentials created
"oort_credentials_delete"	Cisco Identity Intelligence OIDC credentials deleted
"oort_integration_create"	Cisco Identity Intelligence integration created
"oort_integration_delete"	Cisco Identity Intelligence integration deleted
"oort_integration_update"	Cisco Identity Intelligence integration updated
"oort_tenant_create"	Cisco Identity Intelligence tenant created
"oort_tenant_delete"	Cisco Identity Intelligence tenant deleted
"oort_tenant_update"	Cisco Identity Intelligence tenant updated
"oort_webhook_register"	Cisco Identity Intelligence webhook registered
"openldap_connection_config_download"	Admin downloaded authproxy.cfg file for an OpenLDAP sync connection
"openldap_sync_begin"	OpenLDAP Sync started
"openldap_sync_by_user_begin"	OpenLDAP Sync of individual user started
"openldap_sync_by_user_finish"	OpenLDAP Sync of individual user completed
"openldap_sync_config_download"	OpenLDAP Sync configuration download
"openldap_sync_failed"	OpenLDAP Sync failed
"openldap_sync_finish"	OpenLDAP Sync completed
"phone_associate"	Associated phone with user
"phone_create"	Added phone
"phone_delete"	Deleted phone
"phone_disassociate"	Disassociated phone from user
"phone_update"	Modified phone
"policy_bulk_delete"	Bulk deleted policies
"policy_create"	Added a custom policy
"policy_delete"	Deleted a custom policy
"policy_update"	Modified a policy
"push_verify_approved"	Verification push approved
"push_verify_failed"	Verification push failed
"pwl_initial_activation"	Admin accepted Duo Passwordless terms
"redirect_to_child_account"	Redirected to child account
"regen_mobile"	Generated a new Duo Mobile Activation Code
"regen_sms"	Generated new SMS passcodes and sent them to a phone
"reparenting_request_response"	Responded to reparenting request
"resend_enroll_codes"	Resent all enrollment links
"reset_admin_auth_attempts"	Reset an admin's authentication attempt failure count
"revoke_session"	Refresh token session revoked

"send_enroll_code"	Sent an enrollment link
"ssp_policy_enforcement_disabled"	Disabled policy for self-service portal
"ssp_policy_enforcement_enabled"	Enabled policy for self-service portal
"subscription_update"	Modified subscription
"triage_security_event"	Security Event triaged
"u2ftoken_create"	Created U2F token
"u2ftoken_delete"	Deleted U2F token
"update_admin_factor_restrictions"	Modified admin factor restrictions
"update_customers_vat_info"	Updated VAT information
"update_firmographics"	Updated firmographics
"update_passport_configuration"	Updated Passport configuration
"update_risk_profile""	Updated Risk Profile
"updated_risk_profile"	Admin modified Trust Monitor risk profile
"user_bulk_activate"	Bulk mobile activation sent
"user_bulk_enroll"	Bulk enrollment
"user_create"	Added user
"user_delete"	Deleted user
"user_import"	Imported users
"user_lockout_cleared"	Unlocked user account
"user_pending_delete"	Marked user for deletion
"user_restore"	Restored deleted user
"user_update"	Modified user
"webauthncredential_create"	Created a WebAuthn credential
"webauthncredential_delete"	Deleted a WebAuthn credential
"webauthncredential_rename"	Renamed a WebAuthn credential
"whitelist_trusted_endpoint"	Endpoint removed from deny list

`description`	String detailing what changed, either as free-form text or serialized JSON. When the description contains JSON it may be either a serialized object or a serialized array of objects. Each key in the object(s) corresponds to a property that was changed. This JSON is intended only to summarize the change, not to be de-serialized. The first example below is for a "user_update" action. The object that changed was a user whose Duo username is "jsmith". The change saved new values for the user's "notes" and "realname" fields, overwriting the previous values if any were set. They correspond to the similarly named fields in the Modify User call in the Admin API and the User Details page in the Duo Admin Panel. The second example shows an "admin_login_error" action. The administrator's login attempt failed because the admin attempted to use SSO but, as indicated by the "error" in the description, SAML login is disabled for administrators on that account.
`isotimestamp`	ISO8601 timestamp of the event.
`object`	The object that was acted on. For example: "jsmith" (for users), "(555) 713-6275 x456" (for phones), or "HOTP 8-digit 123456" (for tokens).
`timestamp`	An integer indicating the Unix timestamp of the event.

`username`	The full name of the administrator who performed the action in the Duo Admin Panel. If the action was performed with the API this will be "API". Automatic actions like deletion of inactive users have "System" for the username. Changes synchronized from Directory Sync will have a username of the form (example) "AD Sync: name of directory."

EXAMPLE RESPONSES


```
{
  "stat": "OK",
  "response": [{
      "action": "user_update",
      "description": "{\"notes\": \"Joe asked for their nickname to be displayed instead of Josep
      "isotimestamp": "2020-01-24T15:09:42+00:00",
      "object": "jsmith",
      "timestamp": 1579878582,
      "username": "admin"
  }]
}
```

```
{
  "stat": "OK",
  "response": [{
      "action": "admin_login_error",
      "description": "{\"ip_address\": \"10.1.23.116\", \"error\": \"SAML login is disabled\", \"
      "isotimestamp": "2020-01-23T16:18:58+00:00",
      "object": null,
      "timestamp": 1579796338,
      "username": ""
  }]
}
```


Telephony Logs

This API endpoint is currently in Public Preview.

Returns a paged list of telephony log events ranging from the last 180 days up to as recently as two minutes before the API request. The events returned are subject to log retention, if set up in the account, as described here. To fetch all results, call repeatedly with the `next_offset` paging parameter as long as the result metadata has `next_offset` values. Requires "Grant read log" API permission.

There is an intentional two-minute delay in the availability of new telephony events in the API response. Duo operates a large-scale distributed system, and this two-minute buffer period ensures that calls will return consistent results. Querying for results more recent than two minutes will return as empty.

The v2 handler provides new querying capabilities and contextual information for events unavailable in the legacy v1 handler.

We recommend requesting logs no more than once per minute.

`GET` `/admin/v2/logs/telephony`

PARAMETERS

Query Parameter	Required?	Allow Multiple?	Description
`mintime`	**Required**	No	Return records that have a 13 character Unix `timestamp` in milliseconds of `mintime` or later. This value must be strictly less than `maxtime`. Example: `1661022959934`

`maxtime`	**Required**	No	Return records that have a 13 character Unix `timestamp` in milliseconds of `maxtime` or earlier. This value must be strictly greater than `mintime`. Maximum: 180 days Example: `1661022969934`
`limit`	Optional	No	The maximum number of records returned. Default: `100`; Max: `1000`
`next_offset`	Optional	Yes	The offset at which to start record retrieval. This value is provided in the metadata in the form of a 13 character date string in milliseconds and the event `txid`. Both of these values must be provided when used, separated by a comma. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: None Example: `next_offset=1547486297000,5bea1c1e-612c-4f1d-b310-75fd31385b15`
`sort`	Optional	No	The order in which to return records. One of: **Value** / **Description** `ts:asc` — Return logs in chronological order. `ts:desc` — Return logs in reverse chronological order. Default: `ts:desc`

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid or missing parameters or `mintime` is after the `maxtime`.

RESPONSE FORMAT

Key	Value
`items`	List of telephony logs. Information included: `context` — The context under which this telephony event was used (e.g. Administrator Login). `credits` — How many telephony credits this event used. `phone` — The phone number that initiated this event. `telephony_id` — A unique identifier for the telephony event. `ts` — The time at which the telephony event was first recorded, to the best of our ability to record it. This may not always be the exact time the event took place, but should be extremely close. `txid` — A unique identifier that relates to the successful authentication attempt using this telephony event. `type` — The event type. Either "sms" or "phone".

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": {
        "items": [
            {
                "context": "enrollment",
                "credits": 1,
                "phone": "+12125556707",
                "telephony_id": "220f89ff-bff8-4466-b6cb-b7787940ce68",
                "ts": "2023-03-21T22:34:49.466370+00:00",
                "txid": "2f5d34d3-053f-422c-9dd4-77a5d58706b1",
                "type": "sms"
            },
            {
                "context": "authentication",
                "credits": 2,
                "phone": "+17345551311",
                "telephony_id": "60799fee-f08f-4ba8-971f-4e53b3473e9a",
                "ts": "2023-01-26T20:54:12.573580+00:00",
                "txid": "373bd5f3-1e42-4a5d-aefa-b33ae278fac8",
                "type": "phone"
            },
            {
                "context": "administrator login",
                "credits": 0,
                "phone": "+13135559542",
                "telephony_id": "5bf1a860-fe39-49e3-be29-217659663a74",
                "ts": "2022-10-25T16:07:45.304526+00:00",
                "txid": "fb0c129b-f994-4d3d-953b-c3e764272eb7",
                "type": "sms"
            }
        ],
        "metadata": {
            "next_offset": "1666714065304,5bf1a860-fe39-49e3-be29-217659663a74",
            "total_objects": 3
        }
    }
}
```


Telephony Logs (Legacy v1)

The v2 telephony logs endpoint provides new querying capabilities and contextual information for events unavailable in the legacy v1 handler. Consider migrating to the new handler.

Returns a list of telephony log events. Only the 1000 earliest events will be returned; you may need to call this multiple times with `mintime` to page through the entire log. Requires "Grant read log" API permission.

We recommend requesting logs no more than once per minute.

`GET` `/admin/v1/logs/telephony`

PARAMETERS

Parameter	Required?	Description
`mintime`	Optional	Only return records that have a Unix `timestamp` in seconds after `mintime`. This can help to avoid fetching records that have already been retrieved.

RESPONSE CODES

Response	Meaning

200	Success.

RESPONSE FORMAT

Key	Value
`context`	How this telephony event was initiated. One of: "administrator login", "authentication", "enrollment", or "verify".
`credits`	How many telephony credits this event cost.
`isotimestamp`	ISO8601 timestamp of the event.
`phone`	The phone number that initiated this event.
`timestamp`	An integer indicating the Unix timestamp of the event.
`type`	The event type. Either "sms" or "phone".

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": [{
      "context": "authentication",
      "credits": 1,
      "isotimestamp": "2020-03-20T15:38:12+00:00",
      "phone": "+15035550100",
      "timestamp": 1584718692,
      "type": "sms"
  }]
}
```

Offline Enrollment Logs

Returns a list of Duo Authentication for Windows Logon offline enrollment events ranging from the last 180 days up to as recently as two minutes before the API request. There is an intentional two minute delay in availability of new authentications in the API response. Duo operates a large scale distributed system, and this two minute buffer period ensures that calls will return consistent results. Querying for results more recent than two minutes will return as empty. Requires "Grant read log" API permission.

The 1000 earliest events will be returned; you may need to call this multiple times with `mintime` to page through the entire log. Note that more or fewer than 1000 events may be returned depending on how many actual events exist for the specified `mintime`.

We recommend requesting logs no more than once per minute.

```
GET /admin/v1/logs/offline_enrollment
```

PARAMETERS

Parameter	Required?	Description
`mintime`	Optional	Only return records that have a Unix `timestamp` in seconds of `mintime` or later. Use `mintime+1` to avoid receiving duplicate data.

RESPONSE CODES

Response	Meaning
200	Success.

RESPONSE FORMAT

Key	Value
`action`	The offline enrollment operation. One of "o2fa_user_provisioned", "o2fa_user_deprovisioned", or "o2fa_user_reenrolled".
`description`	Information about the Duo Windows Logon client system as reported by the application. `user_agent` — The Duo Windows Logon application version information and the Windows OS version and platform information. `hostname` — The host name of the system where Duo Windows Logon is installed. `factor` — The type of authenticator used for offline access. One of "duo_otp" or "security_key".
`isotimestamp`	ISO8601 timestamp of the event.
`object`	The Duo Windows Logon integration's `name`.
`timestamp`	An integer indicating the Unix timestamp of the event.
`username`	The Duo username.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": [{
      "action": "o2fa_user_provisioned",
      "description": "{\"user_agent\": \"DuoCredProv/4.0.6.413 (Windows NT 6.3.9600; x64; Server)
      "isotimestamp": "2019-08-30T16:10:05+00:00",
      "object": "Acme Laptop Windows Logon",
      "timestamp": 1567181405,
      "username": "narroway"
  }]
}
```

Trust Monitor

This information is available to Duo Premier and Duo Advantage plan customers.

Retrieve Events

Returns a paged list of events surfaced by Trust Monitor from the last 180 days. To fetch all results, call repeatedly with the next_offset paging parameter as long as the result metadata has next_offset values. Requires "Grant read log" API permission.

We recommend requesting Trust Monitor events no more than once per minute.

```
GET /admin/v1/trust_monitor/events
```

PARAMETERS

Use the paging parameters to change the number of results shown in a response or to retrieve additional results. See Response Paging for details.

Paging Parameter	Required?	Description
`limit`	Optional	The maximum number of records returned. Default: `50`; Max: `200`

`offset`	Optional	The offset from `0` at which to start record retrieval. When used with "limit", the handler will return "limit" records starting at the *n-th* record, where *n* is the offset. Default: `0`

Parameter	Required?	Description
`mintime`	**Required**	Return records that have a 13 character Unix `timestamp` in milliseconds of `mintime` or later. This value must be strictly less than `maxtime`.
`maxtime`	**Required**	Return records that have a 13 character Unix `timestamp` in milliseconds of `maxtime` or earlier. This value must be strictly greater than `mintime`.
`formatter`	Optional	This parameter is currently in Public Preview. Specifies the log format. If omitted, logs are returned in the default format. One of: **Value** / **Description** `0` — Returns logs in the default structure shown in this documentation. `1` — Returns logs in Open Cybersecurity Schema Framework Detection finding class format (v1.2). To learn more about OCSF, please refer to the official OCSF documentation.
`type`	Optional	The type of security event. **Type** / **Description** `auth` — Return security events that are denied anomalous authentications. `bypass_status_enabled` — Return security events that are bypass status enabled. `device_registration` — Return security events that are device registrations.

RESPONSE CODES

Response	Meaning
200	Success. Returns a list of security events.
400	Invalid or missing parameters.

RESPONSE FORMAT

Key	Value
`events`	Array of events that match the Parameters. The information returned for each event includes: `bypass_status_enabled` — An integer indicating the Unix timestamp in milliseconds when bypass status was enabled for the user or group. Returned for events with `type=bypass_status`. `enabled_by` — The application or the administrator that enabled bypass status. Returned for events with `type=bypass_status`.

`enabled_for`	The user or group with bypass status. Returned for events with `type=bypass_status`.
`explanations`	An array of objects describing why Trust Monitor surfaced the event. The `summary` provides additional details. Events with `type=auth`: The `type` value will be one of: `GRANTED_AUTH`, `NEW_COUNTRY_CODE`, `NEW_DEVICE`, `NEW_FACTOR`, `NEW_NETBLOCK`, `UNREALISTIC_GEOVELOCITY`, `UNUSUAL_COUNTRY_CODE`, `UNUSUAL_DEVICE`, `UNUSUAL_FACTOR`, `UNUSUAL_NETBLOCK`, `UNUSUAL_TIME_OF_DAY`, or `USER_MARKED_FRAUD`. Events with `type=device_registration`: The `type` value will be one of: `REGISTER_INACTIVE_USER`, `REGISTER_OS_OUTDATED`, `REGISTER_UNLOCK`, or `REGISTER_TAMPERED`.
`from_common_netblock`	A boolean describing if this event was created from a common IP netblock. Either `true` or `false`. Returned for events with `type=auth`.
`from_new_user`	A boolean describing if this event was created for a new user. Either `true` or `false`. Returned for events with `type=auth` or `type=device_registration`.
`low_risk_ip`	A boolean describing if this event was created from an IP address identified in the Risk Profile configuration as a low risk IP address. Either `true` or `false`. Returned for events with `type=auth`.
`priority_event`	A boolean describing if the event matches the Risk Profile configuration. Either `true` or `false`.
`priority_reasons`	An array of objects describing how the event matches the Trust Monitor Risk Profile configuration. Each object contains: `type` — The type of priority reason for the event's match. `label` — The label of the priority reason for the event's match. Returned for events with `type=auth` or `type=device_registration`.
`sekey`	The unique identifier for this event as a 20 character string. This is unique across all different event types.
`state`	A string describing the state of the event. One of `state` new or `state` processed.
`state_updated_timestamp`	An integer indicating the Unix timestamp in milliseconds of the last change to the state of the event.
`surfaced_auth`	An object which represents the actual authentication. See the Authentication Logs response format for authentication event details. Returned for events with `type=auth`.

`surfaced_timestamp`	An integer indicating the Unix timestamp in milliseconds when the event was surfaced by Trust Monitor.
`triaged_as_interesting`	A boolean describing if this event was triaged as being interesting or not interesting. Either `true` or `false`.
`triage_event_uri`	A string representing the URI of the security event, which a Duo administrator can use to view and process the surfaced event in the Duo Admin Panel. Returned for events with `type=auth`.
`type`	The type of event, as a string. One of `auth`, `bypass_status`, or `device_registration`.

EXAMPLE RESPONSE

Authentication Event:


```
{
    "stat": "OK",
    "response": {
        "events": [
                        {
                "explanations": [
                    {
                        "summary": "amanda_tucker has not logged in from this location recently."
                        "type": "NEW_COUNTRY_CODE"
                    },
                    {
                        "summary": "amanda_tucker has not logged in from this IP recently.",
                        "type": "NEW_NETBLOCK"
                    },
                    {
                        "summary": "amanda_tucker has not accessed this application recently.",
                        "type": "NEW_IKEY"
                    }
                ],
                "from_common_netblock": true,
                "from_new_user": false,
                "low_risk_ip": false,
                "priority_event": true,
                "priority_reasons": [
                    {
                        "label": "CN",
                        "type": "country"
                    }
                ],
                "sekey": "SEDOR9BP00L23C6YUH5",
                "state": "new",
                "state_updated_timestamp": null,
                "surfaced_auth": {
                    "access_device": {
                        "browser": "Chrome",
                        "browser_version": "86.0.4240.198",
                        "epkey": "EP18JX1A10AB102M2T2X",
                        "flash_version": null,
                        "hostname": null,
                        "ip": "17.88.232.83",
```

```
            "is_encryption_enabled": "unknown",
            "is_firewall_enabled": "unknown",
            "is_password_set": "unknown",
            "java_version": null,
            "location": {
                "city": "Shanghai",
                "country": "China",
                "state": "Shanghai"
            },
            "os": "Windows",
            "os_version": "10",
            "security_agents": "unknown"
        },
        "alias": "unknown",
        "application": {
            "key": "DIUD2X62LHMPDP00LXS3",
            "name": "Microsoft Azure Active Directory"
        },
        "auth_device": {
            "ip": null,
            "key": null,
            "location": {
                "city": null,
                "country": null,
                "state": null
            },
            "name": null
        },
        "email": "",
        "event_type": null,
        "factor": "not_available",
        "isotimestamp": "2020-11-17T03:19:13.092+00:00",
        "ood_software": "",
        "reason": "location_restricted",
        "result": "denied",
        "timestamp": 1605583153,
        "trusted_endpoint_status": null,
        "txid": "436694ad-467c-4aed-b048-8ad--f58e04c",
        "user": {
            "groups": [
                "crazy"
            ],
            "key": "DUN73JE5M92DP00L4ZYS",
            "name": "amanda_tucker"
        }
    },
    "surfaced_timestamp": 1605602911680,
    "triage_event_uri": "https://admin-xxxxxxxx.duosecurity.com/trust-monitor?sekey=S
    "triaged_as_interesting": false,
    "type": "auth"
}
],
"metadata": {
    "next_offset": "31229"
}
},
}
```


Bypass Status Enabled Event:


```
{
    "stat": "OK",
```

```
    "response": {
        "events": [
                        {
                "bypass_status_enabled": 1604337058989,
                "enabled_by": {
                    "key": "DEWGH6P00LT2R0I60UI",
                    "name": "Ellery Munson"
                },
                "enabled_for": {
                    "key": "DUN73JE5M92DP00L4ZYS",
                    "name": "amanda_tucker"
                },
                "priority_event": true,
                "priority_reasons": [],
                "sekey": "SEDOR9BP00L23C6YUH5",
                "state": "new",
                "state_updated_timestamp": null,
                "surfaced_timestamp": 1605602911680,
                "triaged_as_interesting": false,
                "type": "bypass_status"
            }
        ],
        "metadata": {}
    },
}
```


Device Registration Event:


```
{
    "stat": "OK",
    "response": {
        "events": [
                        {
                "explanations": [
                    {
                        "summary": "The registered device has an out-of-date version of the opera
                        "type": "REGISTER_OS_OUTDATED"
                    }
                ],
                "from_new_user": false,
                "priority_event": false,
                "priority_reasons": [],
                "sekey": "SEDOR9BP00L23C6YUH7",
                "state": "new",
                "state_updated_timestamp": null,
                "surfaced_timestamp": 1675893605269,
                "triaged_as_interesting": false,
                "type": "device_registration"
            }
        ],
        "metadata": {}
    },
}
```


Settings

Retrieve Settings

Returns global Duo settings. These settings can also be viewed and set in the Duo Admin Panel. Requires "Grant settings" API permission.

`GET` `/admin/v1/settings`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success. Returns settings.

RESPONSE FORMAT

<table>
<tr><th>Key</th><th>Value</th></tr>
<tr><td><code>caller_id</code></td><td>Automated calls will appear to come from this number. This does not apply to text messages.</td></tr>
<tr><td><code>duo_mobile_otp_type</code></td><td>The one-time passcode setting for Duo Mobile. One of:
<table>
<tr><th>Configuration</th><th>Description</th></tr>
<tr><td><code>HOTP Only</code></td><td>Only HOTP codes are generated in Duo Mobile for all end users. Duo-protected applications accept only HOTP codes.</td></tr>
<tr><td><code>Groups may use TOTP</code></td><td>Only certain user groups have TOTP codes generated in Duo Mobile, all other users have HOTP codes generated. To see which groups, log in to the Duo Admin Panel and navigate to the settings page. Both TOTP and HOTP codes are accepted for Duo-protected applications.</td></tr>
<tr><td><code>TOTP and HOTP</code></td><td>Duo Mobile 4.49.0 and newer generate TOTP codes only. Older Duo Mobile versions continue generating HOTP codes. Both TOTP and HOTP codes are accepted for Duo-protected applications.</td></tr>
<tr><td><code>TOTP Only</code></td><td>Duo Mobile 4.49.0 and newer generate TOTP codes only. Older Duo Mobile versions continue generating HOTP codes, but these passcodes will not be accepted. Only TOTP codes are accepted for Duo-protected applications and the Duo Admin Panel.</td></tr>
</table>
See the Passcodes setting for more information.</td></tr>
<tr><td><code>email_activity_notification_enabled</code></td><td>If <code>true</code>, users will receive an email notification when an authentication device is added or removed. If set to <code>false</code>, no email notifications are sent in these situations.</td></tr>
<tr><td><code>enrollment_universal_prompt_enabled</code></td><td>If <code>true</code>, the enrollment experience will show the Universal Prompt. If <code>false</code>, the enrollment experience will show the traditional Duo Prompt.</td></tr>
<tr><td><code>fraud_email</code></td><td>The email address to be notified when a user reports a fraudulent authentication attempt or is locked out due to failed authentication attempts. All administrators will be notified if this is not set.</td></tr>
<tr><td><code>fraud_email_enabled</code></td><td>If <code>true</code>, emailed notifications of user-reported fraudulent authentication attempts and user lockouts due to failed authentication are sent to the email address defined for fraud_email, or to all administrators if fraud_email is not defined. If set to <code>false</code>, no fraud alert emails are sent.</td></tr>
<tr><td><code>global_ssp_policy_enforced</code></td><td>If <code>true</code>, a policy set by an administrator is enforced for users trying to access the self-service portal. If set to <code>false</code>, the policy to access the self-service portal will be determined by the destination application policy.</td></tr>
<tr><td><code>helpdesk_bypass</code></td><td>Grants permission for administrators with the Help Desk role to generate bypass codes for users. One of <code>allow</code> (default value), <code>limit</code>, or <code>deny</code>.</td></tr>
<tr><td><code>helpdesk_bypass_expiration</code></td><td>Integer specifying a default expiration for bypass codes generated by Help Desk admins, in minutes. If not set, Help Desk admins may change bypass code expiration from the default 60 minutes after creation if <code>helpdesk_bypass</code> is set to <code>allow</code>.</td></tr>
</table>

`helpdesk_can_send_enroll_email`	Permits Help Desk administrators to send or resend enrollment emails to users. One of `true` or `false` (default).
`helpdesk_message`	Legacy parameter; no effect if specified. Use the Retrieve Custom Messaging endpoint.
`inactive_user_expiration`	Users will be automatically deleted if they are inactive (no successful logins) for a this amount of days.
`keypress_confirm`	The key for users to press to authenticate, or empty if any key should be pressed to authenticate.
`keypress_fraud`	The key for users to press to report fraud, or empty if any key should be pressed to authenticate.
`language`	The language used in the traditional Duo browser-based user authentication prompt. One of: "EN", "DE", "FR". Default: "EN"
`lockout_expire_duration`	If non-zero, an integer indicating the time in minutes until a locked-out user's status reverts to "Active". If `null` or `0`, a user remains locked out until their status is manually changed (By an admin or API call). Minimum: `5` minutes. Maximum: `30000` minutes.
`lockout_threshold`	The number of consecutive failed authentication attempts before the user's status is set to "Locked Out" and the user is denied access.
`log_retention_days`	When set, log entries older than the specified number of days are purged. Logs retained indefinitely if `null`. Note that the log retention setting does not change the 180 day limitation for viewing and retrieving log information in the Duo Admin Panel, exported reports, or via this API. Minimum: `1` day. Maximum: `365` days.
`minimum_password_length`	An integer indicating the minimum number of characters that an administrator's Duo Admin Panel password must contain. This is only enforced on password creation and reset; existing passwords will not be invalidated. Default: `12`.
`mobile_otp_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`name`	The customer name.
`password_requires_lower_alpha`	If `true`, administrator passwords will be required to contain a lower case alphabetic character. If `false`, administrator passwords will not be required to contain a lower case alphabetic character. This is only enforced on password creation and reset; existing passwords will not be invalidated. Default: `false`.
`password_requires_numeric`	If `true`, administrator passwords will be required to contain a numeric character. If `false`, administrator passwords will not be required to contain a numeric character. This is only enforced on password creation and reset; existing passwords will not be invalidated. Default: `false`.
`password_requires_special`	If `true`, administrator passwords will be required to contain a special (non-alphanumeric) character. If `false`, administrator passwords will not be required to contain a special (non-alphanumeric) character. This is only enforced on password creation and reset; existing passwords will not be invalidated. Default: `false`.
`password_requires_upper_alpha`	If `true`, administrator passwords will be required to contain an upper case alphabetic character. If `false`, administrator passwords will not be required to contain an upper case alphabetic character. This is only enforced on password creation and reset; existing passwords will not be invalidated. Default: `false`.
`push_activity_notification_enabled`	If `true`, users will receive a Duo Mobile notification when an authentication device is added or removed. If set to `false`, no email notifications are sent in these situations.
`push_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`sms_batch`	An integer that indicates how many passcodes to send at one time, up to 10.
`sms_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.

`sms_expiration`	The time in minutes to expire and invalidate SMS passcodes, up to 16,777,215.
`sms_message`	Description sent with every batch of SMS passcodes.
`sms_refresh`	If `1`, a new set of SMS passcodes will automatically be sent after the last one is used. If `0`, a new set will not be sent.
`telephony_warning_min`	An integer indicating the number of telephony credits at which an alert will be sent for low credits.
`timezone`	This is the timezone used when displaying timestamps in the Duo Admin Panel.
`unenrolled_user_lockout_threshold`	If non-zero, this is the number of days users can be unenrolled for before they are put into a locked out status. If `0`, then users will not be put into a locked out status if they are unenrolled for any given period of time. Default value is `0`.
`user_managers_can_put_users_in_bypass`	Permits User Manager administrators to apply "Bypass" status to users. One of `true` (default) or `false`.
`user_telephony_cost_max`	An integer indicating the maximum number of telephony credits a user may consume in a single authentication event. This excludes Duo administrators authenticating to the Duo administration panel. Default: `20`.
`voice_enabled`	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
      "caller_id": "+15035551000",
      "duo_mobile_otp_type": "TOTP and HOTP",
      "email_activity_notification_enabled": false,
      "enrollment_universal_prompt_enabled": true,
      "fraud_email": "example@example.com",
      "fraud_email_enabled": true,
      "global_ssp_policy_enforced": true;
      "helpdesk_bypass": "allow",
      "helpdesk_bypass_expiration": 0,
      "helpdesk_can_send_enroll_email": false,
      "helpdesk_message": "",
      "inactive_user_expiration": 30,
      "keypress_confirm": "#",
      "keypress_fraud": "*",
      "language": "EN",
      "lockout_expire_duration": null,
      "lockout_threshold": 10,
      "log_retention_days": null,
      "minimum_password_length": 12,
      "mobile_otp_enabled": false,
      "name": "Acme Corp",
      "password_requires_lower_alpha": true,
      "password_requires_numeric": true,
      "password_requires_special": false,
      "password_requires_upper_alpha": true,
      "push_enabled": false,
      "req_fips_passcodes_android": false,
      "security_checkup_enabled": 1,
      "sms_batch": 1,
      "sms_enabled": false,
      "sms_expiration": 10,
      "sms_message": "SMS passcodes",
```

```
        "sms_refresh": 0,
        "telephony_warning_min": 0,
        "timezone": "UTC",
        "user_managers_can_put_users_in_bypass": true,
        "user_telephony_cost_max": 20.0,
        "voice_enabled": false,
    }
}
```

Modify Settings

Change global Duo settings. Requires "Grant settings" API permission.

`POST` `/admin/v1/settings`

PARAMETERS

<table>
<tr><th>Parameter</th><th>Required?</th><th>Description</th></tr>
<tr><td><code>caller_id</code></td><td>Optional</td><td>Automated calls will appear to come from this number. This does not apply to text messages. Customizing this number may cause telephony providers to flag your number as fraudulent and result in failed user authentications.</td></tr>
<tr><td><code>duo_mobile_otp_type</code></td><td>Optional</td><td>The one-time passcode setting for Duo Mobile. One of:<table><tr><th>Configuration</th><th>Description</th></tr><tr><td><code>hotp_only</code></td><td>Only HOTP codes are generated in Duo Mobile for all end users. Duo-protected applications accept only HOTP codes.</td></tr><tr><td><code>groups_may_use_totp</code></td><td>Only certain user groups have TOTP codes generated in Duo Mobile, all other users have HOTP codes generated. To configure which groups, log in to the Duo Admin Panel and navigate to the settings page. Both TOTP and HOTP codes are accepted for Duo-protected applications.</td></tr><tr><td><code>totp_and_hotp</code></td><td>Duo Mobile 4.49.0 and newer generate TOTP codes only. Older Duo Mobile versions continue generating HOTP codes. Both TOTP and HOTP codes are accepted for Duo-protected applications.</td></tr><tr><td><code>totp_only</code></td><td>Duo Mobile 4.49.0 and newer generate TOTP codes only. Older Duo Mobile versions continue generating HOTP codes, but these passcodes will not be accepted. Only TOTP codes are accepted for Duo-protected applications and the Duo Admin Panel. This option is not reversible! Only enable this once you are sure all users and administrators are running Duo Mobile 4.49.0 or newer!</td></tr></table>See the Passcodes setting for more information.</td></tr>
<tr><td><code>email_activity_notification_enabled</code></td><td>Optional</td><td>If <code>true</code>, users will receive an email notification when an authentication device is added or removed. If set to</td></tr>
</table>

		`false`, no email notifications are sent in these situations. Default value is `false`.
`enrollment_universal_prompt_enabled`	Optional	If `true`, the enrollment experience will show the Universal Prompt. If `false`, the enrollment experience will show the traditional Duo Prompt. Default: show the Universal Prompt (for new customers as of July 2022).
`fraud_email`	Optional	The email address to be notified when a user reports a fraudulent authentication attempt or is locked out due to failed authentication attempts, or empty for all administrators will be notified. If fraud_email is set to a specific email address and fraud_email_enabled is set to `false`, the specific email address value is cleared.
`fraud_email_enabled`	Optional	Set to `true` to enable fraudulent authentication notification emails. False disables the fraud email functionality. If fraud_email is set to a specific email address and fraud_email_enabled is set to `false`, the specific email address value is cleared.
`global_ssp_policy_enforced`	Optional	If `true`, a policy set by an administrator is enforced for users trying to access the self-service portal. If set to `false`, the policy to access the self-service portal will be determined by the destination application policy. Default value is `true`.
`helpdesk_bypass`	Optional	Grants permission for administrators with the Help Desk role to generate bypass codes for users. The default value `allow` permits unrestricted generation of bypass codes, `limit` plus a value for `helpdesk_bypass_expiration` allows Help Desk admins to generate bypass codes with a preset expirtation, and `deny` prevents Help Desk admins from generating any bypass codes.
`helpdesk_bypass_expiration`	Optional	Integer specifying a default expiration for bypass codes generated by Help Desk admins, in minutes. If not set, Help Desk admins may change bypass code expiration from the default 60 minutes after creation if `helpdesk_bypass` is set to `allow`. If specifying a value, also set `helpdesk_bypass` to `limit`.
`helpdesk_can_send_enroll_email`	Optional	Permits Help Desk administrators to send or resend enrollment emails to users. Set to `true` to allow sending of enrollment emails. Default value is `false`.
`helpdesk_message`	Optional	Legacy parameter; no effect if specified. Use the Modify Custom Messaging endpoint.
`inactive_user_expiration`	Optional	Users will be automatically deleted if they are inactive (no successful logins) for this number of days. Minimum: `30` Maximum: `365`
`keypress_confirm`	Optional	The key for users to press to authenticate, or empty if any key should be pressed to authenticate. If this is empty, `keypress_fraud` must be as well.
`keypress_fraud`	Optional	The key for users to report fraud, or empty if any key should be pressed to authenticate. If this is empty, `keypress_confirm` must be as well.
`language`	Optional	Sets the language used in the browser-based user authentication prompt. One of: "EN", "DE", "FR". Default: "EN"
`lockout_expire_duration`	Optional	If non-zero, the time in minutes until a locked-out user's status reverts to "Active". If `0`, a user remains locked out until their status is manually changed (By an admin or API call). Minimum: `5` Maximum: `30000`
`lockout_threshold`	Optional	The number of consecutive failed authentication attempts before the user's status is set to "Locked Out" and the user is denied access. Default is `10` attempts. Minimum: `1` Maximum: `9999`
`log_retention_days`	Optional	When set, log entries older than the specified number of days are purged. Logs retained indefinitely if `null`. Note that the log retention setting does not change the 180 day limitation for viewing and retrieving log information in the Duo Admin Panel, exported reports, or via this API. Default: `null` (no retention limit). Minimum: `1` day. Maximum: `365` days.

`minimum_password_length`	Optional	The minimum number of characters that an administrator's Duo Admin Panel password must contain. This is only enforced on password creation and reset; existing passwords will not be invalidated. Default: `12`. Minimum: `12` Maximum: `100`
`mobile_otp_enabled`	Optional	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`name`	Optional	Sets the customer name.
`password_requires_lower_alpha`	Optional	If `true`, administrator passwords will be required to contain a lower case alphabetic character. If `false`, administrator passwords will not be required to contain a lower case alphabetic character. This is only enforced on password creation and reset; existing passwords will not be invalidated. Default: `false`.
`password_requires_numeric`	Optional	If `true`, administrator passwords will be required to contain a numeric character. If `false`, administrator passwords will not be required to contain a numeric character. This is only enforced on password creation and reset; existing passwords will not be invalidated. Default: `false`.
`password_requires_special`	Optional	If `true`, administrator passwords will be required to contain a special (non-alphanumeric) character. If `false`, administrator passwords will not be required to contain a special (non-alphanumeric) character. This is only enforced on password creation and reset; existing passwords will not be invalidated. Default: `false`.
`password_requires_upper_alpha`	Optional	If `true`, administrator passwords will be required to contain an upper case alphabetic character. If `false`, administrator passwords will not be required to contain an upper case alphabetic character. This is only enforced on password creation and reset; existing passwords will not be invalidated. Default: `false`.
`push_activity_notification_enabled`	Optional	If `true`, users will receive a Duo Mobile notification when an authentication device is added or removed. If set to `false`, no email notifications are sent in these situations. Default value is `false`.
`push_enabled`	Optional	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`sms_batch`	Optional	The number of passcodes to send at one time, up to 10.
`sms_enabled`	Optional	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.
`sms_expiration`	Optional	The time in minutes to expire and invalidate SMS passcodes, or empty if they should not expire.
`sms_message`	Optional	Description sent with every batch of SMS passcodes.
`sms_refresh`	Optional	If `1`, a new set of SMS passcodes will automatically be sent after the last one is used. If `0`, a new set will not be sent.
`telephony_warning_min`	Optional	Configure a alert to be sent when the account has fewer than this many telephony credits remaining.
`timezone`	Optional	This is the timezone used when displaying timestamps in the Duo Admin Panel. Timezones must be entries in the IANA Time Zone Database, for example, "US/Eastern", "Australia/Darwin", "GMT".
`u2f_enabled`	Optional	Legacy parameter; no effect if specified. Use Duo Authentication Method policies to configure this setting.

`unenrolled_user_lockout_threshold`	Optional	If non-zero, this is the number of days users can be unenrolled for before they are put into a locked out status. If `0`, then users will not be put into a locked out status if they are unenrolled for any given period of time. Default value is `0`.
`user_managers_can_put_users_in_bypass`	Optional	Permits User Manager administrators to apply "Bypass" status to users. Set to `false` to prevent User Managers from applying "Bypass" status. Default value is `true`.
`user_telephony_cost_max`	Optional	The maximum number of telephony credits a user may consume in a single authentication event. This excludes Duo administrators authenticating to the Duo administration panel. If you know the countries from which your users expect to authenticate with phone callback we recommend adjusting this down from the default to match the most expensive expected country to help avoid misuse, using the values from the Telephony Credits documentation. Default: `20`.
`voice_enabled`	Optional	Legacy parameter; no effect if specified and always returns `false`. Use Duo Authentication Method policies to configure this setting.

RESPONSE CODES

Response	Meaning
200	The settings were modified successfully. The settings object is also returned (see Retrieve Settings).
400	Invalid or missing parameters. For example, `inactive_user_expiration` out of bounds.

RESPONSE FORMAT

Same as Retrieve Settings.

EXAMPLE RESPONSE

Same as Retrieve Settings.

Retrieve Logo

Returns the custom logo displayed in the Duo authentication prompt and Duo Mobile. Requires "Grant settings API permission.

This logo customization is superseded by Custom Branding for Duo Premier, Advantage, and Essentials plan customers. Migrate to the new custom branding endpoint for increased functionality. This endpoint is deprecated and will stop working in a future update.

```
GET /admin/v1/logo
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success. Returns PNG data.
404	No logo currently configured.

RESPONSE FORMAT

If there are no errors, a PNG image is returned instead of JSON and the Content-Type header is image/png.

Modify Logo

Returns the custom logo displayed in the Duo authentication prompt and Duo Mobile. This logo is sent to devices when they enroll with the mobile app. Currently enrolled devices must be re-activated to receive the new logo. Requires "Grant settings" API permission.

This logo customization is superseded by Custom Branding for Duo Premier, Advantage, and Essentials plan customers. Migrate to the new custom branding endpoint for increased functionality. This endpoint is deprecated and will stop working in a future update.

`POST` `/admin/v1/logo`

PARAMETERS

Parameter	Required?	Description
`logo`	**Required**	Base-64 encoded PNG image data. The logo image must be in PNG format and not exceed 500 by 500 pixels and 200 KB. We recommend a 304 by 304 pixel logo image with a transparent background for the best results.

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid or missing parameters or PNG data.
600	Incorrect PNG base64 encoding.

RESPONSE FORMAT

Empty string.

Delete Logo

Remove the logo from the Duo authentication prompt and future activation of Duo Mobile. Currently enrolled devices must be re-activated to remove the logo. Requires "Grant settings" API permission.

This logo customization is superseded by Custom Branding for Duo Premier, Advantage, and Essentials plan customers. Migrate to the new custom branding endpoint for increased functionality. This endpoint is deprecated and will stop working in a future update. Logo updates made to this endpoint have no effect.

`DELETE` `/admin/v1/logo`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The logo was deleted or did not exist.

RESPONSE FORMAT

Empty string.

Custom Branding

Custom branding is available to Duo Premier, Duo Advantage, and Duo Essentials plan customers.

Retrieve Live Custom Branding

Returns effective custom branding settings. These settings can also be viewed and set in the Duo Admin Panel. Requires "Grant settings" API permission.

```
GET  /admin/v1/branding
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success. Returns live branding.

RESPONSE FORMAT

Key	Value
`background_img`	A base64 encoded background image in PNG format. Shown in Duo SSO and Duo Universal Prompt.
`card_accent_color`	A CSS hex color shown as the hash symbol (#) followed by three or six hexadecimal digits, which represents the colored line appearing at the top of the interactive user interface. Shown in Duo SSO and Universal Prompt.
`logo`	A base64 encoded logo image in PNG format. Shown in Duo SSO, Duo Universal Prompt, and traditional prompt.
`page_background_color`	A CSS hex color shown as the hash symbol (#) followed by three or six hexadecimal digits, which represents the color appearing below the user interface and beneath any transparent background image. Shown in Duo SSO and Universal Prompt.
`powered_by_duo`	If `true`, Duo SSO, Duo Universal Prompt, and traditional prompt show the "Secured by Duo" branding. Otherwise, `false`.
`sso_custom_username_label`	A string that is displayed for the custom SSO Login Label.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "background_image": "iVBORw0KGgoAAAANSUhEUgAABQAAAAMgCAYAAAB8mM/7AAABrmlUWHRYTUw6Y29tLmFkb2JlI
    "card_accent_color": "#144A94",
    "logo": "iVBORw0KGgoAAAANSUhEUgAAARQAAAEwCAYAAAB2TY5ZAAAACXBIWXMAAAwHAAAMBwF2myPLAAA55mlUWHRY
    "page_background_color": "#437BC6",
    "powered_by_duo": true,
    "sso_custom_username_label": "Username"
  }
}
```


Modify Live Custom Branding

Change effective custom branding settings. These settings can also be viewed and set in the Duo Admin Panel. Requires "Grant settings" API permission.

```
POST  /admin/v1/branding
```

PARAMETERS

Parameter	Required?	Description
`background_img`	Optional	A base64 encoded background image in PNG format, with maximum size less than 3MB and dimensions between 12 by 12 pixels and 3840 by

		2160 pixels. Shown in Duo SSO and Duo Universal Prompt.
`card_accent_color`	Optional	A CSS hex color shown as the hash symbol (#) followed by three or six hexadecimal digits, which represents the colored line appearing at the top of the interactive user interface. Shown in Duo SSO and Universal Prompt.
`logo`	Optional	A base64 encoded logo image in PNG format, with maximum size less than 200KB and dimensions between 12 by 12 pixels and 500 by 500 pixels. Shown in Duo SSO, Duo Universal Prompt, and traditional prompt.
`page_background_color`	Optional	A CSS hex color shown as the hash symbol (#) followed by three or six hexadecimal digits, which represents the color appearing behind the user interface and any transparent background image. Shown in Duo SSO and Universal Prompt.
`powered_by_duo`	Optional	If `true`, Duo SSO, Duo Universal Prompt, and traditional prompt show the "Secured by Duo" branding. Otherwise, `false`.
`sso_custom_username_label`	Optional	Specify the string that is displayed for the custom SSO Login Label. Can be `Username`, `Email Address`, or a custom string. The custom string can only contain letters and numbers (maximum length 100 characters). No effect unless Duo SSO is enabled and configured.

RESPONSE CODES

Response	Meaning
200	The live branding settings were modified successfully. The settings objects are also returned (see Retrieve Live Custom Branding).
400	Invalid or missing parameters. For example, `card_accent_color` an invalid HTML hex code.
600	Incorrect PNG base64 encoding of logo or background images.

RESPONSE FORMAT

Same as Retrieve Live Custom Branding.

EXAMPLE RESPONSE

Same as Retrieve Live Custom Branding.

Retrieve Draft Custom Branding

Returns saved draft custom branding settings. These settings can also be viewed and set in the Duo Admin Panel. Requires "Grant settings" API permission.

`GET` `/admin/v1/branding/draft`

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success. Returns draft branding.

RESPONSE FORMAT

Key	Value
`background_img`	A base64 encoded background image in PNG format. Shown in Duo SSO and Duo Universal Prompt.
`card_accent_color`	A CSS hex color shown as the hash symbol (#) followed by three or six hexadecimal digits, which represents the colored line appearing at the top of the interactive user interface. Shown in Duo SSO and Universal Prompt.

`draft_user`	A list of test users who see draft branding (if configured) instead of live branding when using Duo SSO or Duo Universal Prompt.See Retrieve Users for descriptions of the response fields.
`logo`	A base64 encoded logo image in PNG format. Shown in Duo SSO, Duo Universal Prompt, and traditional prompt.
`page_background_color`	A CSS hex color shown as the hash symbol (#) followed by three or six hexadecimal digits, which represents the color appearing behind the user interface and any transparent background image. Shown in Duo SSO and Universal Prompt.
`powered_by_duo`	If `true`, Duo SSO, Duo Universal Prompt, and traditional prompt show the "Secured by Duo" branding. Otherwise, `false`.
`sso_custom_username_label`	A string that is displayed for the custom SSO Login Label.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "background_image": "iVBORw0KGgoAAAANSUhEUgAABQAAAAMgCAYAAAB8mM/7AAABrmlUWHRYTUw6Y29tLmFkb2Jl
    "card_accent_color": "#144A94",
    "draft_users": [{
        "alias1": "joe.smith",
        "alias2": "jsmith@example.com",
        "alias3": null,
        "alias4": null,
        "aliases": {
            "alias1": "joe.smith",
            "alias2": "jsmith@example.com"
        },
        "created": 1384275337,
        "email": "jsmith@example.com",
        "external_id": "1a2345b6-7cd8-9e0f-g1hi-23j45kl6m789",
        "firstname": "",
        "is_enrolled": true,
        "last_directory_sync": 1384275337,
        "last_login": 1514922986,
        "lastname": "",
        "notes": "",
        "realname": "Joe Smith",
        "status": "active",
        "user_id": "DU3RP9I2WOC59VZX672N",
        "username": "jsmith",
    }],
    "logo": "iVBORw0KGgoAAAANSUhEUgAAARQAAAEwCAYAAAB2TY5ZAAAACXBIWXMAAAwHAAAMBwF2myPLAAA55mlUWHRY
    "page_background_color": "#437BC6",
    "powered_by_duo": true,
    "sso_custom_username_label": "Username"
  }
}
```


Modify Draft Custom Branding

Change draft custom branding settings. These settings can also be viewed and set in the Duo Admin Panel. Requires "Grant settings" API permission.

`POST` `/admin/v1/branding/draft`

PARAMETERS

Parameter	Required?	Description

`background_img`	Optional	A base64 encoded background image in PNG format, with maximum size less than 3MB and dimensions between 12 by 12 pixels and 3840 by 2160 pixels. Shown in Duo SSO and Duo Universal Prompt.
`card_accent_color`	Optional	A CSS hex color shown as the hash symbol (#) followed by three or six hexadecimal digits, which represents the colored line appearing at the top of the interactive user interface. Shown in Duo SSO and Universal Prompt.
`logo`	Optional	A base64 encoded logo image in PNG format, with maximum size less than 200KB and dimensions between 12 by 12 pixels and 500 by 500 pixels. Shown in Duo SSO, Duo Universal Prompt, and traditional prompt.
`page_background_color`	Optional	A CSS hex color shown as the hash symbol (#) followed by three or six hexadecimal digits, which represents the color appearing below the user interface and beneath any transparent background image. Shown in Duo SSO and Universal Prompt.
`powered_by_duo`	Optional	If `true`, Duo SSO, Duo Universal Prompt, and traditional prompt show the "Secured by Duo" branding. Otherwise, `false`.
`sso_custom_username_label`	Optional	Specify the string that is displayed for the custom SSO Login Label. Can be `Username`, `Email Address`, or a custom string (maximum length 100 characters). The custom string can only contain letters and numbers. No effect unless Duo SSO is enabled and configured.
`user_ids`	Optional	A comma separated list of user IDs that will see saved draft branding in Duo SSO and Duo Universal Prompt.

RESPONSE CODES

Response	Meaning
200	The draft branding settings were modified successfully. The settings objects are also returned (see Retrieve Live Custom Branding).
400	Invalid or missing parameters. For example, `card_accent_color` an invalid HTML hex code.
600	Incorrect PNG base64 encoding of logo or background images.

RESPONSE FORMAT

Same as Retrieve Draft Custom Branding.

EXAMPLE RESPONSE

Same as Retrieve Draft Custom Branding.

Add Draft Custom Branding User by ID

Add a single user with ID `user_id` to the list of draft branding test users. Requires "Grant settings" API permission.

```
POST  /admin/v1/branding/draft/users/[user_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The user was added to the draft branding user list successfully. The draft branding object is also returned (see Retrieve Draft Custom Branding).
404	No user was found with the given `user_id`.

RESPONSE FORMAT

Same as Retrieve Draft Custom Branding.

EXAMPLE RESPONSE

Same as Retrieve Draft Custom Branding.

Remove Draft Custom Branding User by ID

Remove a single user with ID `user_id` from the list of draft branding test users. Requires "Grant settings" API permission.

```
DELETE  /admin/v1/branding/draft/users/[user_id]
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	The user was removed from the draft branding user list successfully. The draft branding object is also returned (see Retrieve Draft Custom Branding).
404	No user was found with the given `user_id`.

RESPONSE FORMAT

Same as Retrieve Draft Custom Branding.

EXAMPLE RESPONSE

Same as Retrieve Draft Custom Branding.

Publish Draft Custom Branding as Live Custom Branding

Replaces the current live custom branding with the draft custom branding for all users and then removes the draft branding. Requires "Grant settings" API permission.

```
POST  /admin/v1/branding/draft/publish
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.

RESPONSE FORMAT

Same as Retrieve Live Custom Branding.

EXAMPLE RESPONSE

Same as Retrieve Live Custom Branding.

Retrieve Custom Messaging

Returns effective custom messaging settings, shown to users in the Universal Prompt. These settings can also be viewed and set in the Duo Admin Panel. Supersedes the helpdesk_message Settings parameter. Requires "Grant settings" API permission.

```
GET /admin/v1/branding/custom_messaging
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success. Returns custom messaging.

RESPONSE FORMAT

Key	Value
`help_links`	A comma-separated list of up to two custom external links shown to users in the Universal Prompt. Must begin with `http` or `https`.
`help_text_by_locale`	Customized text string associated with the specified locale. The user's browser's preferred language settings determine which language to show in the Universal Prompt. The first locale and message text in the list matches the default language specified in global Settings and is also shown in the traditional web prompt and in Duo Desktop. `help_text`: Text shown to users in the Universal Prompt; up to 200 characters. No support for hyperlinks. `locale`: The language of the help text. One of: `en_US` (English), `ca_ES` (Catalan), `cs_CZ` (Czech), `da_DK` (Danish), `de_DE` (German), `es_ES` (Spanish - Spain), `es_419` (Spanish - Latin America), `fi_FI` (Finnish), `fr_FR` (French), `hi_IN` (Hindi), `id_ID` (Indonesian), `it_IT` (Italian), `ja_JP` (Japanese), `ko_KR` (Korean), `nb_NO` (Norwegian - Bokmål), `nl_NL` (Dutch), `pl_PL` (Polish), `pt_BR` (Portuguese - Brazil), `pt_PT` (Portuguese - Portugal), `sv_SE` (Swedish), `th_TH` (Thai), `tr_TR` (Turkish), `vi_VN` (Vietnamese), `zh_Hans_CN` (Chinese - Simplified), or `zh_Hant_TW` (Chinese - Traditional).

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": {
      "help_links": [
           "https://helpme.acme.corp"
      ],
      "help_text_by_locale": [
           {
              "help_text": "Contact the Acme Service Desk at 888-555-5310.",
              "locale": "en_US"
           },
          {
              "help_text": "Contactez le Service Desk Acme au 888-555-5310.",
              "locale": "fr_FR"
          }
      ]
```

```
    }
}
```

Modify Custom Messaging

Updates current custom messaging settings, shown to users in the Universal Prompt. These settings can also be viewed and set in the Duo Admin Panel. Supersedes the helpdesk_message Settings parameter. Requires "Grant settings" API permission.

```
POST  /admin/v1/branding/custom_messaging
```

PARAMETERS

Parameter	Required?	Description
`help_links`	Optional	A comma-separated list of up to two custom external links shown to users in the Universal Prompt. Each URL must begin with `http://` or `https://`.
`help_text`	Optional	Customized text string associated with the specified locale. The user's browser's preferred language settings determine which language to show in the Universal Prompt. The first locale and message text in the list matches the default language specified in global Settings and is also shown in the traditional web prompt and in Duo Desktop. Up to 200 characters. No support for hyperlinks.
`locale`	**Required** if `help_text` is specified. Otherwise, optional.	The language of the help text. One of: `en_US` (English), `ca_ES` (Catalan), `cs_CZ` (Czech), `de_DE` (German), `es_ES` (Spanish - Spain), `es_419` (Spanish - Latin America), `fi_FI` (Finnish), `fr_FR` (French), `hi_IN` (Hindi), `id_ID` (Indonesian), `it_IT` (Italian), `ja_JP` (Japanese), `ko_KR` (Korean), `nb_NO` (Norwegian - Bokmål), `nl_NL` (Dutch), `pl_PL` (Polish), `pt_BR` (Portuguese - Brazil), `pt_PT` (Portuguese - Portugal), `sv_SE` (Swedish), `th_TH` (Thai), `tr_TR` (Turkish), `vi_VN` (Vietnamese), `zh_Hans_CN` (Chinese - Simplified), or `zh_Hant_TW` (Chinese - Traditional).

RESPONSE CODES

Response	Meaning
200	The custom messaging settings were updated. The settings objects are also returned (see Retrieve Custom Messaging).
400	Invalid or missing parameters. For example, `help_links` that do not start with HTTP/HTTPS.

RESPONSE FORMAT

Same as Retrieve Custom Messaging.

EXAMPLE RESPONSE

Same as Retrieve Custom Messaging.

Account Info

Retrieve Summary

Returns a summary of account utilization information. Requires "Grant read information" API permission.

```
GET  /admin/v1/info/summary
```

PARAMETERS

This API endpoint has no parameters.

RESPONSE CODES

Response	Meaning
200	Success.

RESPONSE FORMAT

Key	Value
`admin_count`	Current number of admins in the account.
`edition`	Current edition (including trials) of the account.
`integration_count`	Current number of integrations in the account.
`telephony_credits_remaining`	Current total number of telephony credits available in the account. This is the sum of all types of telephony credits.
`user_count`	Current number of users in the account.
`user_pending_deletion_count`	Current number of users pending deletion from the account (seen in the Admin Panel's Trash view).

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": {
    "admin_count": 21,
    "edition": "Duo Premier",
    "integration_count": 44,
    "telephony_credits_remaining": 960,
    "user_count": 862,
    "user_pending_deletion_count": 9
  }
}
```

Telephony Credits Used Report

Retrieve the number of telephony credits used in a given time period. Requires "Grant read information" API permission.

If the specified time period is too long you may need to call this multiple times with `mintime` and sum the results.

`GET` `/admin/v1/info/telephony_credits_used`

PARAMETERS

Parameter	Required?	Description
maxtime	Optional	Limit report to events before this Unix timestamp. Defaults to the current time.
mintime	Optional	Limit report to events after this Unix timestamp. Defaults to thirty days before `maxtime`.

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid or missing parameters or `mintime` is after the `maxtime`.

RESPONSE FORMAT

Key	Value
`mintime`	An integer indicating the Unix timestamp in seconds for the beginning of the report period.
`maxtime`	An integer indicating the Unix timestamp in seconds for the end of the report period.
`telephony_credits_used`	An integer indicating the number of telephony credits consumed during the specified time period.

EXAMPLE RESPONSE

```
{
  "stat": "OK",
  "response": {
    "maxtime": 1352880416,
    "mintime": 1350288416,
    "telephony_credits_used": 30
  }
}
```

Authentication Attempts Report

Retrieve counts of authentication attempts for a given time period (not to exceed 180 days), broken down by result. Requires "Grant read information" API permission.

`GET` `/admin/v1/info/authentication_attempts`

PARAMETERS

Parameter	Required?	Description
maxtime	Optional	Limit report to events before this Unix timestamp. Defaults to the current time.
mintime	Optional	Limit report to events after this Unix timestamp. Defaults to thirty days before `maxtime`.

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid or missing parameters, `mintime` is outside the 180 day retention window, or `mintime` is after the `maxtime`.

RESPONSE FORMAT

Key	Value
`mintime`	An integer indicating the Unix timestamp in seconds for the beginning of the report period.
`maxtime`	An integer indicating the Unix timestamp in seconds for the end of the report period.
`authentication_attempts`	An integer indicating the number of authentication attempts during the specified time period, broken down by result:

Result	Meaning
"ERROR"	An unexpected failure prevented authentication (for example, an invalid telephone number).
"FAILURE"	Authentication was denied.
"FRAUD"	The attempt ended with a report of fraudulent activity.
"SUCCESS"	Authentication was allowed.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "authentication_attempts": {
      "ERROR": 0,
      "FAILURE": 1,
      "FRAUD": 1,
      "SUCCESS": 50
    },
    "maxtime": 1352880416,
    "mintime": 1350288416
  }
}
```


Users with Authentication Attempts Report

Retrieve counts of users with authentication attempts for a given time period (not to exceed 180 days), broken down by result. Each count is the number of users who had at least one authentication attempt ending with that result. Requires "Grant read information" API permission.

`GET` `/admin/v1/info/user_authentication_attempts`

PARAMETERS

Parameter	Required?	Description
maxtime	Optional	Limit report to events before this Unix timestamp. Defaults to the current time.
mintime	Optional	Limit report to events after this Unix timestamp. Defaults to thirty days before `maxtime`.

RESPONSE CODES

Response	Meaning
200	Success.
400	Invalid or missing parameters, `mintime` is outside the 180 day retention window, or `mintime` is after the `maxtime`.

RESPONSE FORMAT

Key	Value
`mintime`	An integer indicating the Unix timestamp in seconds for the beginning of the report period.
`maxtime`	An integer indicating the Unix timestamp in seconds for the end of the report period.
`user_authentication_attempts`	An integer indicating the number of users with authentication attempts during the specified time period, broken down by result. Refer to Authentication Attempts Report for a list of result types and their meanings.

EXAMPLE RESPONSE


```
{
  "stat": "OK",
  "response": {
    "maxtime": 1352880416,
    "mintime": 1350288416,
    "user_authentication_attempts": {
      "ERROR": 0,
      "FAILURE": 1,
      "FRAUD": 1,
      "SUCCESS": 10
```

```
      }
    }
}
```

Troubleshooting

Need some help? Take a look at our Admin API Knowledge Base articles or Community discussions. For further assistance, contact Support.