

[Skip to main content](#)

- [New](#)
- [Sofia Fuhrman](#)

Dropbox for HTTP Developers

[Overview](#)[User Endpoints](#)[Business Endpoints](#)

Dropbox API v2

The Dropbox API allows developers to work with files in Dropbox, including advanced functionality like full-text search, thumbnails, and sharing. The [Dropbox API explorer](#) is the easiest way to get started making API calls.

Request and response formats

In general, the Dropbox API uses HTTP POST requests with JSON arguments and JSON responses. Request authentication is via OAuth 2.0 using the Authorization request header or authorization URL parameter.

The `.tag` field in an object identifies the subtype of a struct or selected member of a union.

When specifying a Void member of a union, you may supply just the member string in place of the entire tagged union object. For example, when supplying a `WriteMode`, you can supply just `"mode": "add"` instead of `"mode": {"tag": "add"}{}`. This shorthand is not allowed for non-Void members. For example, the following is not allowed for a `WriteMode`, as `update` is not a Void member: `"mode": "update"`.

RPC endpoints

These endpoints accept arguments as JSON in the request body and return results as JSON in the response body. RPC endpoints are on the `api.dropboxapi.com` domain.

Content-upload endpoints

These endpoints accept file content in the request body, so their arguments are instead passed as JSON in the `Dropbox-API-Arg` request header or `arg` URL parameter. These endpoints are on the `content.dropboxapi.com` domain.

Content-download endpoints

As with content-upload endpoints, arguments are passed in the `Dropbox-API-Arg` request header or `arg` URL parameter. The response body contains file content, so the result will appear as JSON in the `Dropbox-API-Result` response header. These endpoints are also on the `content.dropboxapi.com` domain.

These endpoints also support HTTP GET along with ETag-based caching (`If-None-Match`) and HTTP range requests.

For information on how to properly encode the JSON, see the [JSON encoding page](#).

Browser-based JavaScript and CORS pre-flight requests

When browser-based JavaScript code makes a cross-site HTTP request, the browser must sometimes send a "pre-flight" check to make sure the server allows cross-site requests. You can avoid the extra round-trip by ensuring your request meets the CORS definition of a "simple cross-site request".

- Use URL parameters `arg` and `authorization` instead of HTTP headers `Dropbox-API-Arg` and `Authorization`.
- Set the `Content-Type` to `"text/plain; charset=dropbox-cors-hack"` instead of `"application/json"` or `"application/octet-stream"`.
- Always set the URL parameter `reject_cors_preflight=true`. This makes it easier to catch cases where your code is unintentionally triggering a pre-flight check.

Date format

All dates in the API use UTC and are strings in the [ISO 8601 "combined date and time representation" format](#):

2015-05-15T15:50:38Z

Path formats

Paths are relative to an application's root (either an app folder or the root of a user's Dropbox, depending on the app's [access type](#)). The empty string ("") represents the root folder. All other paths must start with a slash (e.g. `"/hello/world.txt"`). Paths may not end with a slash or whitespace. For other path restrictions, refer to the [help center](#).

Every file and folder in Dropbox also has an ID (e.g. `"id:abc123xyz"`) that can be obtained from any endpoint that returns metadata. These IDs are case-sensitive, so they should always be stored with their case preserved, and always compared in a case-sensitive manner. Some endpoints, as noted in the individual endpoint

[/count](#)[/create](#)[/delete](#)[/delete_all_closed](#)[/get](#)[/list](#)[/list_continue](#)[/update](#)[files](#)[/copy](#)[/copy_batch](#)[/copy_batch/check](#)[/copy_reference/get](#)[/copy_reference/save](#)[/create_folder](#)[/create_folder_batch](#)[/create_folder_batch/check](#)[/delete](#)[/delete_batch](#)[/delete_batch/check](#)[/download](#)[/download_zip](#)[/export](#)[/get_file_lock_batch](#)[/get_metadata](#)[/get_preview](#)[/get_temporary_link](#)[/get_temporary_upload_link](#)[/get_thumbnail](#)[/get_thumbnail_batch](#)[/list_folder](#)[/list_folder_continue](#)[/list_folder_get_latest_cursor](#)[/list_folder_longpoll](#)[/list_revisions](#)[/lock_file_batch](#)[/move](#)[/move_batch](#)[/move_batch_check](#)[/permanently_delete](#)[/restore](#)[/save_url](#)[/save_url_check_job_status](#)[/search](#)[/search_continue](#)[/unlock_file_batch](#)[/upload](#)[/upload_session_append](#)[/upload_session_finish](#)[/upload_session_finish_batch](#)[/upload_session_finish_batch_check](#)[/upload_session_start](#)[sharing](#)[/add_file_member](#)[/add_folder_member](#)[/check_job_status](#)[/check_remove_member_job_status](#)



For endpoints that accept performing actions on behalf of a team administrator using the [Dropbox-API-Select-Admin header](#), files may be referenced using a namespace-relative path (e.g. "ns:123456/cupcake.png"). In this case, the namespace ID, "123456", would be the `shared_folder_id` or `team_folder_id` of the shared folder or the team folder containing the file or folder, and the path, "/cupcake.png", would be the logical path to the content relative to its shared folder or team folder container.

Path case insensitivity

Like in Dropbox itself, paths in the Dropbox API are case-insensitive, meaning that /A/B/c.txt is the same file as /a/b/C.txt and is the same file as /a/B/c.txt.

This can cause problems for apps that store file metadata from users in case-sensitive databases (such as SQLite or Postgres). Case insensitive collations should be used when storing Dropbox path metadata in such databases. Alternatively, developers need to make sure their query operators are explicitly case insensitive.

Also, while Dropbox is case-insensitive, it makes efforts to be case-preserving. `Metadata.name` will contain the correct case. `Metadata.path_display` usually will contain the correct case, but sometimes only in the last path component. If your app needs the correct case for all path components, it can get it from the `Metadata.name` or last path component of each relevant `Metadata.path_display` entry.

Authorization

Dropbox supports [OAuth 2.0](#) for authorizing API requests. Find out more in our [OAuth guide](#). Authorized requests to the API should use an `Authorization` header with the value `Bearer <TOKEN>`, where `<TOKEN>` is an access token obtained through the OAuth flow.

Note: OAuth is an authorization protocol, not an authentication protocol. Dropbox should not be used as an identity provider.

`/oauth2/authorize`

DESCRIPTION This starts the OAuth 2.0 authorization flow. This isn't an API call—it's the web page that lets the user sign in to Dropbox and authorize your app. After the user decides whether or not to authorize your app, they will be redirected to the URI specified by `redirect_uri`.

OAuth 2.0 supports two authorization flows:

- The code flow returns a code via the `redirect_uri` callback which should then be converted into a bearer token using the [/oauth2/token call](#). This is the recommended flow for apps that are running on a server.
- The token or implicit grant flow returns the bearer token via the `redirect_uri` callback, rather than requiring your app to make a second call to a server. This is useful for pure client-side apps, such as mobile apps or JavaScript-based apps.

For more information on the two flows, see [Section 1.3 of the OAuth 2 spec](#).

Your app should send the user to this app authorization page in their system browser. If the user isn't already signed in to the Dropbox website, they will be prompted to do so on this web page. This web page should not be displayed in a web-view. This is in order to maintain compatibility with the website and to comply with [Google's policy against processing their OAuth flow inside a web-view](#), to support users who sign in to Dropbox using their Google accounts. Learn about [the dropbox.com system requirements](#).

URL STRUCTURE

`https://www.dropbox.com/oauth2/authorize`

Note: This is the only step that requires an endpoint on `www.dropbox.com`. All other API requests are done via `api.dropboxapi.com`, `content.dropboxapi.com`, or `notify.dropboxapi.com`.

METHOD

GET

EXAMPLE

`https://www.dropbox.com/oauth2/authorize?client_id=<APP_KEY>&response_type=code`

PARAMETERS

response_type *String* The grant type requested, either `token` or `code`.

client_id *String* The app's key, found in the [App Console](#).

redirect_uri *String?* Where to redirect the user after authorization has completed. This must be the exact URI registered in the [App Console](#); even 'localhost' must be listed if it is used for testing. All redirect URIs must be HTTPS except for localhost URIs. A redirect URI is required for the token flow, but optional for the



state *String?* Up to 500 bytes of arbitrary data that will be passed back to your redirect URI. This parameter should be used to protect against cross-site request forgery (CSRF). See Sections [4.4.1.8](#) and [4.4.2.5](#) of the OAuth 2.0 threat model spec.

require_role *String?* If this parameter is specified, the user will be asked to authorize with a particular type of Dropbox account, either work for a team account or personal for a personal account. Your app should still verify the type of Dropbox account after authorization since the user could modify or remove the require_role parameter.

force_reapprove Boolean? Whether or not to force the user to approve the app again if they've already done so. If false (default), a user who has already approved the application may be automatically redirected to the URI specified by `redirect_uri`. If true, the user will not be automatically redirected and will have to approve the app again.

disable_signup Boolean? When true (default is false) users will not be able to sign up for a Dropbox account via the authorization page. Instead, the authorization page will show a link to the Dropbox iOS app in the App Store. This is only intended for use when necessary for compliance with App Store policies.

locale *String?* If the locale specified is a supported language, Dropbox will direct users to a translated version of the authorization website. Locale tags should be IETF language tags.

force_reauthentication Boolean? When true (default is false) users will be signed out if they are currently signed in. This will make sure the user is brought to a page where they can create a new account or sign in to another account. This should only be used when there is a definite reason to believe that the user needs to sign in to a new or different account.

RETURNS Because /oauth2/authorize is a website, there is no direct return value. However, after the user authorizes your app, they will be sent to your redirect URI. The type of response varies based on the `response_type`.

Code flow

These parameters are passed in the query string (after the ? in the URL):

`code` *String* The authorization code, which can be used to attain a bearer token by calling [`/oauth2/token`](#).

state *String* The state content, if any, originally passed to /oauth2/authorize.

Sample response

[REDIRECT URI]?code=ABCDEFG&state=[STATE]

Token flow

These parameters are passed in the URL fragment (after the # in the URL).

Note: as fragments, these parameters can be modified by the user and must not be trusted server-side. If any of these fields are being used server-side, please consider using the **Code flow**, or alternatively using the fields returned from [/get current account](#) instead.

`access_token` *String* A token which can be used to make calls to the Dropbox API.

`token type String` The type of token, which will always be `bearer`.

`account_id` *String* A user's account identifier used by API v2

`team_id` String A team's identifier used by API v2

uid *String* Deprecated. The API v1 user/team identifier. Please use account_id instead, or if using the Dropbox Business API, team_id.

`state String` The state content, if any, originally passed to `/oauth2/authorize`.

Sample response

ERRORS In either flow, if an error occurs, including if the user has chosen not to authorize the app, the following parameters will be included in the redirect URI:

`error` *String* An error code per [Section 4.1.2.1 of the OAuth 2.0 spec](#).

`error_description` *String* A user-friendly description of the error that occurred.



/oauth2/token

DESCRIPTION	This endpoint only applies to apps using the authorization code flow . An app calls this endpoint to acquire a bearer token once the user has authorized the app. Calls to <code>/oauth2/token</code> need to be authenticated using the app's key and secret. These can either be passed as <code>application/x-www-form-urlencoded</code> POST parameters (see parameters below) or via HTTP basic authentication . If basic authentication is used, the app key should be provided as the username, and the app secret should be provided as the password.
-------------	--

URL STRUCTURE

<https://api.dropboxapi.com/oauth2/token>

METHOD POST

```
EXAMPLE curl https://api.dropbox.com/oauth2/token \
    -d code=<AUTHORIZATION_CODE> \
    -d grant_type=authorization_code \
    -d redirect_uri=<REDIRECT_URI> \
    -u <APP_KEY>:<APP_SECRET>
```

PARAMETERS	
	<code>code</code> <i>String</i> The code acquired by directing users to <code>/oauth2/authorize?response_type=code</code> .
	<code>grant_type</code> <i>String</i> The grant type, which must be <code>authorization_code</code> .
	<code>client_id</code> <i>String?</i> If credentials are passed in POST parameters, this parameter should be present and should be the app's key (found in the App Console).
	<code>client_secret</code> <i>String?</i> If credentials are passed in POST parameters, this parameter should be present and should be the app's secret.
	<code>redirect_uri</code> <i>String?</i> Only used to validate that it matches the original <code>/oauth2/authorize</code> , not used to redirect again.

RETURNS A JSON-encoded dictionary including an access token (`access_token`), token type (`token_type`), an API v2 user ID (`account_id`), or if team-linked, an API v2 team ID (`team_id`) instead. The API v1 identifier value (`uid`) is deprecated and should no longer be used. The token type will always be "bearer".

Sample response

Errors

Errors are returned using standard HTTP error code syntax. Depending on the status code, the response body may be in JSON or plaintext.

Errors by status code

Code	Description
400	Bad input parameter. The response body is a plaintext message with more information.
401	Bad or expired token. This can happen if the access token is expired or if the access token has been revoked by Dropbox or the user. To fix this, you should re-authenticate the user.
	The Content-Type of the response is JSON of type <i>AuthError</i>
403	The user or team account doesn't have access to the endpoint or feature.
	The Content-Type of the response is JSON of type <i>AccessError</i>
409	Endpoint-specific error. Look to the JSON response body for the specifics of the error.
429	Your app is making too many requests for the given user or team and is being rate limited. Your app should wait for the number of seconds specified in the "Retry-After" response header before trying again.



5xx An error occurred on the Dropbox servers. Check status.dropbox.com for announcements about Dropbox service issues.

Endpoint-specific errors (409)

The following table describes the top-level JSON object attributes present in the body of 409 responses.

Key	Description
error	A value that conforms to the error data type schema defined in the definition of each route.
error_summary	A string that summarizes the value of the "error" key. It is a concatenation of the hierarchy of union tags that make up the error. This provides a human-readable error string, and can also be used for simple programmatic error handling via prefix matching. To disincentivize exact string matching, we append a random number of "." characters at the end of the string.
user_message	An optional field. If present, it includes a message that can be shown directly to the end user of your app. You should show this message if your app is unprepared to programmatically handle the error returned by an endpoint.

Generate access token

Use the dropdown to select which app to make API calls with. An access token for the chosen app will be generated and inserted into the examples below.

By generating an access token, you will be able to make API calls for your own account without going through the authorization flow. To obtain access tokens for other users, use the standard OAuth flow.

ProfesorRobot ▾

Populate Access Token

account

/set_profile_photo

VERSION 1 ▾

DESCRIPTION Sets a user's profile photo.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/account/set_profile_photo

AUTHENTICATION

[User Authentication](#)

RPC
FORMAT

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/account/set_profile_photo \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"photo\": {\".tag\": \"base64_data\", \"base64_data\": \"SWlhZ2UgZGF0YSBpbBiYXNlNjQtZW5jb2RlZCBieXRLc\"}}
```

PARAMETERS

```
{ "photo": {
    ".tag": "base64_data",
    "base64_data": "SWlhZ2UgZGF0YSBpbBiYXNlNjQtZW5jb2RlZCBieXRLcIy4gTm90IGEgdmFsaWQgZXhhbXBsZS4="
}}
```

SetProfilePhotoArg

photo PhotoSourceArg Image to set as the user's new profile photo.

RETURNS

```
{ "profile_photo_url": "https://dl-web.dropbox.com/account_photo/get/dbaphid%3AAAHWGmIXV3sUu0mBfTz0wPsiqHUpBWv3z"}
```



ERRORS Example: file_type_error

```
{  
    "error_summary": "file_type_error/...",  
    "error": {  
        ".tag": "file_type_error"  
    }  
}
```

Example: file_size_error

```
{  
    "error_summary": "file_size_error/...",  
    "error": {  
        ".tag": "file_size_error"  
    }  
}
```

Example: dimension_error

```
{  
    "error_summary": "dimension_error/...",  
    "error": {  
        ".tag": "dimension_error"  
    }  
}
```

Example: thumbnail_error

```
{  
    "error_summary": "thumbnail_error/...",  
    "error": {  
        ".tag": "thumbnail_error"  
    }  
}
```

Example: transient_error

```
{  
    "error_summary": "transient_error/...",  
    "error": {  
        ".tag": "transient_error"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

SetProfilePhotoError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

file_type_error *Void* File cannot be set as profile photo.

file_size_error *Void* File cannot exceed 10 MB.

dimension_error *Void* Image must be larger than 128 x 128.

thumbnail_error *Void* Image could not be thumbnailized.

transient_error *Void* Temporary infrastructure failure, please retry.

auth

/token/from_oauth1

VERSION

1 ▾

DESCRIPTION Creates an OAuth 2.0 access token from the supplied OAuth 1.0 access token.

URL STRUCTURE

https://api.dropboxapi.com/2/auth/token/from_oauth1

AUTHENTICATION [App Authentication](#)

ENDPOINT
FORMAT

EXAMPLE Get app key and secret for: ProfesorRobot ▾



```
--data "{\"oauth1_token\": \"qievr8hamyg6ndck\", \"oauth1_token_secret\": \"qomoftv0472git7\"}"
```

PARAMETERS

```
{  
    "oauth1_token": "qievr8hamyg6ndck",  
    "oauth1_token_secret": "qomoftv0472git7"  
}
```

TokenFromOAuth1Arg

oauth1_token *String(min_length=1)* The supplied OAuth 1.0 access token.

oauth1_token_secret *String(min_length=1)* The token secret associated with the supplied access token.

RETURNS

```
{  
    "oauth2_token": "9mCrkS7BIdAAAAAAAAAHHS0TsSnpYvKQVtKdBnN5IuzhY0GbLsgTcHgBFKFmFn"  
}
```

TokenFromOAuth1Result

oauth2_token *String(min_length=1)* The OAuth 2.0 token generated from the supplied OAuth 1.0 token.

ERRORS

Example: invalid_oauth1_token_info

```
{  
    "error_summary": "invalid_oauth1_token_info/...",  
    "error": {  
        ".tag": "invalid_oauth1_token_info"  
    }  
}
```

Example: app_id_mismatch

```
{  
    "error_summary": "app_id_mismatch/...",  
    "error": {  
        ".tag": "app_id_mismatch"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

TokenFromOAuth1Error (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

invalid_oauth1_token_info *Void* Part or all of the OAuth 1.0 access token info is invalid.

app_id_mismatch *Void* The authorized app does not match the app associated with the supplied access token.

/token/revoke

VERSION

1 ▾

DESCRIPTION

Disables the access token used to authenticate the call.

URL STRUCTURE

<https://api.dropboxapi.com/2/auth/token/revoke>

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/auth/token/revoke \  
--header "Authorization: Bearer <get access token>"
```

PARAMETERS

No parameters.

RETURNS

No return values.



check

/app

VERSION

1

PREVIEW - may change or disappear without notice

DESCRIPTION

This endpoint performs App Authentication, validating the supplied app key and secret, and returns the supplied string, to allow you to test your code and connection to the Dropbox API. It has no other effect. If you receive an HTTP 200 response with the supplied query, it indicates at least part of the Dropbox API infrastructure is working and that the app key and secret valid.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

<https://api.dropboxapi.com/2/check/app>

AUTHENTICATION

[App Authentication](#)

ENDPOINT
FORMAT

[RPC](#)

EXAMPLE

Get app key and secret for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/check/app \
  --header "Authorization: Basic <get app key and secret>" \
  --header "Content-Type: application/json" \
  --data "{\"query\": \"foo\"}"
```

PARAMETERS

```
{  
  "query": "foo"  
}
```

EchoArg

EchoArg contains the arguments to be sent to the Dropbox servers.

query *String* The string that you'd like to be echoed back to you. The default for this field is "".

RETURNS

```
{  
  "result": "foo"  
}
```

EchoResult

EchoResult contains the result returned from the Dropbox servers.

result *String* If everything worked correctly, this would be the same as query. The default for this field is "".

ERRORS

No errors.

/user

VERSION

1

PREVIEW - may change or disappear without notice

DESCRIPTION

This endpoint performs User Authentication, validating the supplied access token, and returns the supplied string, to allow you to test your code and connection to the Dropbox API. It has no other effect. If you receive an HTTP 200 response with the supplied query, it indicates at least part of the Dropbox API infrastructure is working and that the access token is valid.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

<https://api.dropboxapi.com/2/check/user>

AUTHENTICATION

[User Authentication](#)

ENDPOINT
FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾



```
--data "{\"query\": \"foo\"}"
```

PARAMETERS

```
{  
  "query": "foo"  
}
```

EchoArg

EchoArg contains the arguments to be sent to the Dropbox servers.

query *String* The string that you'd like to be echoed back to you. The default for this field is "".

RETURNS

```
{  
  "result": "foo"  
}
```

EchoResult

EchoResult contains the result returned from the Dropbox servers.

result *String* If everything worked correctly, this would be the same as query. The default for this field is "".

ERRORS

No errors.

contacts**/delete_manual_contacts****VERSION**

1 ▾

DESCRIPTION

Removes all manually added contacts. You'll still keep contacts who are on your team or who you imported. New contacts will be added when you share.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

```
https://api.dropboxapi.com/2/contacts/delete_manual_contacts
```

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/contacts/delete_manual_contacts \  
--header "Authorization: Bearer <get access token>"
```

PARAMETERS

No parameters.

RETURNS

No return values.

ERRORS

No errors.

/delete_manual_contacts_batch**VERSION**

1 ▾

DESCRIPTION

Removes manually added contacts from the given list.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

```
https://api.dropboxapi.com/2/contacts/delete_manual_contacts_batch
```

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾



contacts_batch \

--data "{\"email_addresses\": [\"contactemailaddress1@domain.com\", \"contactemailaddress2@domain.com\"]}"]

PARAMETERS

```
{  
  "email_addresses": [  
    "contactemailaddress1@domain.com",  
    "contactemailaddress2@domain.com"  
  ]  
}
```

DeleteManualContactsArg

email_addresses List of (String(max_length=255, pattern="^['A-Za-z0-9._%+-]+@[A-Za-z0-9.-]*[A-Za-z]{2,15}\$")) List of manually added contacts to be deleted.

RETURNS No return values.

ERRORS DeleteManualContactsError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

contacts_not_found List of (String(max_length=255, pattern="^['A-Za-z0-9._%+-]+@[A-Za-z0-9.-]*[A-Za-z]{2,15}\$")) Can't delete contacts from this list. Make sure the list only has manually added contacts. The deletion was cancelled.

file_properties

This namespace contains helpers for property and template metadata endpoints.

These endpoints enable you to tag arbitrary key/value data to Dropbox files.

The most basic unit in this namespace is the PropertyField. These fields encapsulate the actual key/value data.

Fields are added to a Dropbox file using a PropertyGroup. Property groups contain a reference to a Dropbox file and a PropertyGroupTemplate. Property groups are uniquely identified by the combination of their associated Dropbox file and template.

The PropertyGroupTemplate is a way of restricting the possible key names and value types of the data within a property group. The possible key names and value types are explicitly enumerated using PropertyFieldTemplate objects.

You can think of a property group template as a class definition for a particular key/value metadata object, and the property groups themselves as the instantiations of these objects.

Templates are owned either by a user/app pair or team/app pair. Templates and their associated properties can't be accessed by any app other than the app that created them, and even then, only when the app is linked with the owner of the template (either a user or team).

User-owned templates are accessed via the user-auth file_properties/templates/*_for_user endpoints, while team-owned templates are accessed via the team-auth file_properties/templates/*_for_team endpoints.

Properties associated with either type of template can be accessed via the user-auth properties/* endpoints.

Finally, properties can be accessed from a number of endpoints that return metadata, including `files/get_metadata`, and `files/list_folder`. Properties can also be added during upload, using `files/upload`.

/properties/add

VERSION

1 ▾

DESCRIPTION

Add property groups to a Dropbox file. See [templates/add_for_user](#) or [templates/add_for_team](#) to create new templates.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/file_properties/properties/add

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾



```
--data "{\"path\": \"/my_awesome/word.docx\", \"property_groups\": [{\"template_id\": \"ptid:1a5n2i6d30YEAaaaaaa\"}
```

PARAMETERS

```
{  
    "path": "/my_awesome/word.docx",  
    "property_groups": [  
        {  
            "template_id": "ptid:1a5n2i6d30YEAaaaaaaAYa",  
            "fields": [  
                {  
                    "name": "Security Policy",  
                    "value": "Confidential"  
                }  
            ]  
        }  
    ]  
}
```

AddPropertiesArg

path *String(pattern="/(.|[r\n])*|id:.*|(ns:[0-9]+(\\.*))")* A unique identifier for the file or folder.

property_groups *List of (PropertyGroup)* The property groups which are to be added to a Dropbox file.

RETURNS

No return values.

ERRORS

Example: restricted_content

```
{  
    "error_summary": "restricted_content/...",  
    "error": {  
        ".tag": "restricted_content"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: unsupported_folder

```
{  
    "error_summary": "unsupported_folder/...",  
    "error": {  
        ".tag": "unsupported_folder"  
    }  
}
```

Example: property_field_too_large

```
{  
    "error_summary": "property_field_too_large/...",  
    "error": {  
        ".tag": "property_field_too_large"  
    }  
}
```

Example: does_not_fit_template

```
{  
    "error_summary": "does_not_fit_template/...",  
    "error": {  
        ".tag": "does_not_fit_template"  
    }  
}
```

Example: property_group_already_exists

```
{  
    "error_summary": "property_group_already_exists/...",  
    "error": {  
        ".tag": "property_group_already_exists"  
    }  
}
```

AddPropertiesError (*union*)

The value will be one of the following datatypes:

template_not_found *String(min_length=1, pattern="(|ptid:).*)")* Template does not exist for the given identifier.

restricted_content *Void* You do not have permission to modify this template.

path *LookupError*



property field values is too large.

does_not_fit_template *Void* One or more of the supplied property fields does not conform to the template specifications.

property_group_already_exists *Void* A property group associated with this template and file already exists.

/properties/overwrite

VERSION

1

DESCRIPTION Overwrite property groups associated with a file. This endpoint should be used instead of [properties/update](#) when property groups are being updated via a "snapshot" instead of via a "delta". In other words, this endpoint will delete all omitted fields from a property group, whereas [properties/update](#) will only delete fields that are explicitly marked for deletion.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/file_properties/properties/overwrite

AUTHENTICATION

[User Authentication](#)

ENDPOINT
FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/file_properties/properties/overwrite \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"path\": \"/my_awesome/word.docx\", \"property_groups\": [{\"template_id\": \"ptid:1a5n2i6d30YEAAAAAAA\"}
```

PARAMETERS

```
{  
    "path": "/my_awesome/word.docx",  
    "property_groups": [  
        {  
            "template_id": "ptid:1a5n2i6d30YEAAAAAAAAYa",  
            "fields": [  
                {  
                    "name": "Security Policy",  
                    "value": "Confidential"  
                }  
            ]  
        }  
    ]  
}
```

OverwritePropertyGroupArg

path *String(pattern="/(.|[\\r\\n])*|id:.*|(ns:[0-9]+(/.*)?)"")* A unique identifier for the file or folder.

property_groups *List of (PropertyGroup, min_items=1)* The property groups "snapshot" updates to force apply.

RETURNS

No return values.

ERRORS Example: restricted_content

```
{  
    "error_summary": "restricted_content/...",  
    "error": {  
        ".tag": "restricted_content"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: unsupported_folder

```
{  
    "error_summary": "unsupported_folder/...",  
    "error": {  
        ".tag": "unsupported_folder"  
    }  
}
```



```
    "error_summary": "property_field_too_large/...",
    "error": {
        ".tag": "property_field_too_large"
    }
}
```

Example: does_not_fit_template

```
{
    "error_summary": "does_not_fit_template/...",
    "error": {
        ".tag": "does_not_fit_template"
    }
}
```

InvalidPropertyGroupError (*union*)

The value will be one of the following datatypes:

template_not_found *String(min_length=1, pattern="(|ptid:).*)")*
Template does not exist for the given identifier.

restricted_content *Void* You do not have permission to modify this template.

path *LookupError*

unsupported_folder *Void* This folder cannot be tagged. Tagging folders is not supported for team-owned templates.

property_field_too_large *Void* One or more of the supplied property field values is too large.

does_not_fit_template *Void* One or more of the supplied property fields does not conform to the template specifications.

/properties/remove

VERSION

1

DESCRIPTION Permanently removes the specified property group from the file. To remove specific property field key value pairs, see [properties/update](#). To update a template, see [templates/update_for_user](#) or [templates/update_for_team](#). To remove a template, see [templates/remove_for_user](#) or [templates/remove_for_team](#).

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/file_properties/properties/remove

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/file_properties/properties/remove \
--header "Authorization: Bearer <get access token> " \
--header "Content-Type: application/json" \
--data "{\"path\": \"/my_awesome/word.docx\", \"property_template_ids\": [\"ptid:la5n2i6d30YEAFFFFFFAYa\"]}"
```

PARAMETERS

```
{
    "path": "/my_awesome/word.docx",
    "property_template_ids": [
        "ptid:la5n2i6d30YEAFFFFFFAYa"
    ]
}
```

RemovePropertiesArg

path *String(pattern="/(.|[r\n])*|id:.*|(ns:[0-9]+(/.?)?)"")* A unique identifier for the file or folder.

property_template_ids *List of (String(min_length=1, pattern="(|ptid:).*)")* A list of identifiers for a template created by [templates/add_for_user](#) or [templates/add_for_team](#).

RETURNS

No return values.

ERRORS

Example: restricted_content

```
{
    "error_summary": "restricted_content/...",
    "error": {
        ".tag": "restricted_content"
    }
}
```



```
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

Example: unsupported_folder

```
{
    "error_summary": "unsupported_folder/...",
    "error": {
        ".tag": "unsupported_folder"
    }
}
```

RemovePropertiesError (*union*)

The value will be one of the following datatypes:

template_not_found *String*(*min_length*=1, *pattern*="(|ptid:).*)")

Template does not exist for the given identifier.

restricted_content *Void* You do not have permission to modify this template.

path *LookupError*

unsupported_folder *Void* This folder cannot be tagged. Tagging folders is not supported for team-owned templates.

property_group_lookup *LookUpPropertiesError*

/properties/search

VERSION

1

DESCRIPTION Search across property templates for particular property field values.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/file_properties/properties/search

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/file_properties/properties/search \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"queries\": [{\"query\": \"Confidential\", \"mode\": {\".tag\": \"field_name\", \"field_name\": \"Security\"}}, {\"logical_operator\": \"or_operator\"}], \"template_filter\": \"filter_none\"}
```

PARAMETERS

```
{
    "queries": [
        {
            "query": "Confidential",
            "mode": {
                ".tag": "field_name",
                "field_name": "Security"
            },
            "logical_operator": "or_operator"
        }
    ],
    "template_filter": "filter_none"
}
```

PropertiesSearchArg

queries *List of (PropertiesSearchQuery, min_items=1)* Queries to search.

template_filter *TemplateFilter* Filter results to contain only properties associated with these template IDs. The default for this union is filter_none.

RETURNS

```
{
    "matches": [
        {
            "id": "id:a4ayc_80_0EAAAAAAAAXz",
            "path": "/my_awesome/word.docx",
            "is_deleted": false,
            "property_groups": [
                {
                    "template_id": "ptid:1a5n2i6d30YEAAAAAAAAYa",
                    "fields": [
                        {
                            "name": "Security Policy",
                            "value": "Confidential"
                        }
                    ]
                }
            ]
        }
    ]
}
```



≡

{] }

PropertiesSearchResult

matches *List of (PropertiesSearchMatch)* A list (possibly empty) of matches for the query.

cursor *String(min_length=1)?* Pass the cursor into [properties/search/continue](#) to continue to receive search results. Cursor will be null when there are no more results. This field is optional.

ERRORS

PropertiesSearchError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

property_group_lookup *LookUpPropertiesError*

/properties/search/continue

VERSION

1 ▾

DESCRIPTION

Once a cursor has been retrieved from [properties/search](#), use this to paginate through all search results.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/file_properties/properties/search/continue

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/file_properties/properties/search/continue \
  -header "Authorization: Bearer <get access token>" \
  -header "Content-Type: application/json" \
  -data "{\"cursor\": \"ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu\"}"
```

PARAMETERS

```
{\n    "cursor": "ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu"\n}
```

PropertiesSearchContinueArg

cursor *String(min_length=1)* The cursor returned by your last call to [properties/search](#) or [properties/search/continue](#).

RETURNS

```
{\n    "matches": [\n        {\n            "id": "id:a4ayc_80_0EAAAAAAAAXz",\n            "path": "/my_awesome/word.docx",\n            "is_deleted": false,\n            "property_groups": [\n                {\n                    "template_id": "ptid:la5n2i6d30YEAFFFFFFAYa",\n                    "fields": [\n                        {\n                            "name": "Security Policy",\n                            "value": "Confidential"\n                        }\n                    ]\n                }\n            ]\n        }\n    ]\n}
```

PropertiesSearchResult

matches *List of (PropertiesSearchMatch)* A list (possibly empty) of matches for the query.

cursor *String(min_length=1)?* Pass the cursor into [properties/search/continue](#) to continue to receive search results. Cursor will be null when there are no more results. This field is optional.

ERRORS

Example: reset

```
{\n    "error_summary": "reset/...",\n    "error": {\n
```



≡



Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

PropertiesSearchContinueError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

reset *Void* Indicates that the cursor has been invalidated. Call [properties/search](#) to obtain a new cursor.

/properties/update

VERSION

1

DESCRIPTION Add, update or remove properties associated with the supplied file and templates. This endpoint should be used instead of [properties/overwrite](#) when property groups are being updated via a "delta" instead of via a "snapshot". In other words, this endpoint will not delete any omitted fields from a property group, whereas [properties/overwrite](#) will delete any fields that are omitted from a property group.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

```
https://api.dropboxapi.com/2/file_properties/properties/update
```

AUTHENTICATION

[User Authentication](#)

ENDPOINT
FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/file_properties/properties/update \  
    --header "Authorization: Bearer <get access token>" \  
    --header "Content-Type: application/json" \  
    --data "{\"path\": \"/my_awesome/word.docx\", \"update_property_groups\": [{\"template_id\": \"ptid:la5n2i6d30YF\",
```

PARAMETERS

```
{  
    "path": "/my_awesome/word.docx",  
    "update_property_groups": [  
        {  
            "template_id": "ptid:la5n2i6d30YF",  
            "add_or_update_fields": [  
                {  
                    "name": "Security Policy",  
                    "value": "Confidential"  
                }  
            ],  
            "remove_fields": []  
        }  
    ]  
}
```

UpdatePropertiesArg

path *String(pattern="/(.|[r\n])*|id:.*|(ns:[0-9]+(./*)?)"")* A unique identifier for the file or folder.

update_property_groups *List of (PropertyGroupUpdate)* The property groups "delta" updates to apply.

RETURNS

No return values.

ERRORS Example: restricted_content

```
{  
    "error_summary": "restricted_content/...",  
    "error": {  
        ".tag": "restricted_content"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"
```



≡



Example: unsupported_folder

```
{  
    "error_summary": "unsupported_folder/...",  
    "error": {  
        ".tag": "unsupported_folder"  
    }  
}
```

Example: property_field_too_large

```
{  
    "error_summary": "property_field_too_large/...",  
    "error": {  
        ".tag": "property_field_too_large"  
    }  
}
```

Example: does_not_fit_template

```
{  
    "error_summary": "does_not_fit_template/...",  
    "error": {  
        ".tag": "does_not_fit_template"  
    }  
}
```

UpdatePropertiesError (*union*)

The value will be one of the following datatypes:

template_not_found *String*(min_length=1, pattern="(|ptid:).*)")
Template does not exist for the given identifier.

restricted_content *Void* You do not have permission to modify this template.

path *LookupError*

unsupported_folder *Void* This folder cannot be tagged. Tagging folders is not supported for team-owned templates.

property_field_too_large *Void* One or more of the supplied property field values is too large.

does_not_fit_template *Void* One or more of the supplied property fields does not conform to the template specifications.

property_group_lookup *LookUpPropertiesError*

/templates/add_for_user

VERSION

1 ▾

DESCRIPTION Add a template associated with a user. See [properties/add](#) to add properties to a file. This endpoint can't be called on a team member or admin's behalf.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/file_properties/templates/add_for_user

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/file_properties/templates/add_for_user \  
    --header "Authorization: Bearer <get access token>" \  
    --header "Content-Type: application/json" \  
    --data "{\"name\": \"Security\", \"description\": \"These properties describe how confidential this file or folc" data-bbox="223 759 989 802"/>
```

PARAMETERS

```
{  
    "name": "Security",  
    "description": "These properties describe how confidential this file or folder is.",  
    "fields": [  
        {  
            "name": "Security Policy",  
            "description": "This is the security policy of the file or folder described.\n\nPolicies can be Confidential",  
            "type": "string"  
        }  
    ]  
}
```

AddTemplateArg

name *String* Display name for the template. Template names can be up to 256 bytes.



rty

fields associated with this template. There can be up to 32 properties in a single template.

RETURNS

```
{  
    "template_id": "ptid:1a5n2i6d30YEAFFFFFFAYa"  
}
```

AddTemplateResult

template_id *String(min_length=1, pattern="/|ptid:).*)* An identifier for template added by See [templates/add_for_user](#) or [templates/add_for_team](#).

ERRORS

Example: restricted_content

```
{  
    "error_summary": "restricted_content/...",  
    "error": {  
        ".tag": "restricted_content"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: conflicting_property_names

```
{  
    "error_summary": "conflicting_property_names/...",  
    "error": {  
        ".tag": "conflicting_property_names"  
    }  
}
```

Example: too_many_properties

```
{  
    "error_summary": "too_many_properties/...",  
    "error": {  
        ".tag": "too_many_properties"  
    }  
}
```

Example: too_many_templates

```
{  
    "error_summary": "too_many_templates/...",  
    "error": {  
        ".tag": "too_many_templates"  
    }  
}
```

Example: template_attribute_too_large

```
{  
    "error_summary": "template_attribute_too_large/...",  
    "error": {  
        ".tag": "template_attribute_too_large"  
    }  
}
```

ModifyTemplateError (*union*)

The value will be one of the following datatypes:

template_not_found *String(min_length=1, pattern="/|ptid:).*)*
Template does not exist for the given identifier.

restricted_content *Void* You do not have permission to modify this template.

conflicting_property_names *Void* A property field key with that name already exists in the template.

too_many_properties *Void* There are too many properties in the changed template. The maximum number of properties per template is 32.

too_many_templates *Void* There are too many templates for the team.

template_attribute_too_large *Void* The template name, description or one or more of the property field keys is too large.



DESCRIPTION Get the schema for a specified template. This endpoint can't be called on a team member or admin's behalf.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/file_properties/templates/get_for_user

AUTHENTICATION

User Authentication

ENDPOINT FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/file_properties/templates/get_for_user \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"template_id\": \"ptid:la5n2i6d30YEAAAAAAAAYa\"}"
```

PARAMETERS

```
{  
    "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa"  
}
```

GetTemplateArg

template_id String(min_length=1, pattern="/|ptid:).*") An identifier for template added by route See [templates/add_for_user](#) or [templates/add_for_team](#).

RETURNS

```
{  
    "name": "Security",  
    "description": "These properties describe how confidential this file or folder is.",  
    "fields": [  
        {  
            "name": "Security Policy",  
            "description": "This is the security policy of the file or folder described.\nPolicies can be Confidential, Sensitive, or Public.",  
            "type": {  
                ".tag": "string"  
            }  
        }  
    ]  
}
```

GetTemplateResult

name String Display name for the template. Template names can be up to 256 bytes.

description String Description for the template. Template descriptions can be up to 1024 bytes.

fields List of (PropertyFieldTemplate) Definitions of the property fields associated with this template. There can be up to 32 properties in a single template.

ERRORS

Example: restricted_content

```
{  
    "error_summary": "restricted_content/...",  
    "error": {  
        ".tag": "restricted_content"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

TemplateError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

template_not_found String(min_length=1, pattern="/|ptid:).*") Template does not exist for the given identifier.

restricted_content Void You do not have permission to modify this template.

/templates/list_for_user

VERSION

1 ▾



This endpoint does not support apps with the app folder permission.

URL STRUCTURE

```
https://api.dropboxapi.com/2/file_properties/templates/list_for_user
```

AUTHENTICATION

User Authentication

ENDPOINT FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/file_properties/templates/list_for_user \
--header "Authorization: Bearer <get access token>"
```

PARAMETERS

No parameters.

RETURNS

```
{  
    "template_ids": [  
        "ptid:la5n2i6d30YEAFFFFFFAYa"  
    ]  
}
```

ListTemplateResult

template_ids List of (String(min_length=1, pattern="/|ptid:).*"))

List of identifiers for templates added by See [templates/add_for_user](#) or [templates/add_for_team](#).

ERRORS

Example: restricted_content

```
{  
    "error_summary": "restricted_content/...",  
    "error": {  
        ".tag": "restricted_content"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

TemplateError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

template_not_found String(min_length=1, pattern="/|ptid:).*")

Template does not exist for the given identifier.

restricted_content Void You do not have permission to modify this template.

/templates/remove_for_user

VERSION

1 ▾

DESCRIPTION

Permanently removes the specified template created from [templates/add_for_user](#). All properties associated with the template will also be removed. This action cannot be undone.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

```
https://api.dropboxapi.com/2/file_properties/templates/remove_for_user
```

AUTHENTICATION

User Authentication

ENDPOINT FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/file_properties/templates/remove_for_user \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"template_id\": \"ptid:la5n2i6d30YEAFFFFFFAYa\"}"
```



RemoveTemplateArg

template_id String(min_length=1, pattern="/|ptid:.*") An identifier for a template created by [templates/add_for_user](#) or [templates/add_for_team](#).

RETURNS No return values.

Example: restricted_content

```
{  
    "error_summary": "restricted_content/...",  
    "error": {  
        ".tag": "restricted_content"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

TemplateError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

template_not_found String(min_length=1, pattern="/|ptid:.*")

Template does not exist for the given identifier.

restricted_content Void You do not have permission to modify this template.

/templates/update_for_user

VERSION

1

DESCRIPTION Update a template associated with a user. This route can update the template name, the template description and add optional properties to templates. This endpoint can't be called on a team member or admin's behalf.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/file_properties/templates/update_for_user

AUTHENTICATION

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/file_properties/templates/update_for_user \  
    --header "Authorization: Bearer <get access token>" \  
    --header "Content-Type: application/json" \  
    --data "{\"template_id\": \"ptid:la5n2i6d30YEAAAAAAAAYa\", \"name\": \"New Security Template Name\", \"descripti
```

PARAMETERS

```
{  
    "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",  
    "name": "New Security Template Name",  
    "description": "These properties will describe how confidential this file or folder is.",  
    "add_fields": [  
        {  
            "name": "Security Policy",  
            "description": "This is the security policy of the file or folder described.\nPolicies can be Confidential, Sensitive, or Public.",  
            "type": "string"  
        }  
    ]  
}
```

UpdateTemplateArg

template_id String(min_length=1, pattern="/|ptid:.*") An identifier for template added by See [templates/add_for_user](#) or [templates/add_for_team](#).

name String? A display name for the template. template names can be up to 256 bytes. This field is optional.

description String? Description for the new template. Template descriptions can be up to 1024 bytes. This field is optional.



RETURNS

```
{  
    "template_id": "ptid:la5n2i6d30YEAFFFFFFYea"  
}
```

UpdateTemplateResult

template_id *String(min_length=1, pattern="(/ptid:).*)* An identifier for template added by route See [templates/add_for_user](#) or [templates/add_for_team](#).

ERRORS Example: restricted_content

```
{  
    "error_summary": "restricted_content/...",  
    "error": {  
        ".tag": "restricted_content"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: conflicting_property_names

```
{  
    "error_summary": "conflicting_property_names/...",  
    "error": {  
        ".tag": "conflicting_property_names"  
    }  
}
```

Example: too_many_properties

```
{  
    "error_summary": "too_many_properties/...",  
    "error": {  
        ".tag": "too_many_properties"  
    }  
}
```

Example: too_many_templates

```
{  
    "error_summary": "too_many_templates/...",  
    "error": {  
        ".tag": "too_many_templates"  
    }  
}
```

Example: template_attribute_too_large

```
{  
    "error_summary": "template_attribute_too_large/...",  
    "error": {  
        ".tag": "template_attribute_too_large"  
    }  
}
```

ModifyTemplateError (*union*)

The value will be one of the following datatypes:

template_not_found *String(min_length=1, pattern="(/ptid:).*)*
Template does not exist for the given identifier.

restricted_content *Void* You do not have permission to modify this template.

conflicting_property_names *Void* A property field key with that name already exists in the template.

too_many_properties *Void* There are too many properties in the changed template. The maximum number of properties per template is 32.

too_many_templates *Void* There are too many templates for the team.

template_attribute_too_large *Void* The template name, description or one or more of the property field keys is too large.

file_requests

This namespace contains endpoints and data types for file request operations.



≡

VERSION 1 ▾

DESCRIPTION Returns the total number of file requests owned by this user. Includes both open and closed file requests.

URL STRUCTURE https://api.dropboxapi.com/2/file_requests/count

AUTHENTICATION [User Authentication](#)

ENDPOINT FORMAT [RPC](#)

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/file_requests/count \
--header "Authorization: Bearer <get access token>"
```

PARAMETERS No parameters.

RETURNS

```
{ "file_request_count": 15 }
```

CountFileRequestsResult

Result for [count](#).

file_request_count UInt64 The number file requests owner by this user.

ERRORS Example: disabled_for_team

```
{ "error_summary": "disabled_for_team/...", "error": { ".tag": "disabled_for_team" } }
```

Example: other

```
{ "error_summary": "other/...", "error": { ".tag": "other" } }
```

CountFileRequestsError (*union*)

There was an error counting the file requests. The value will be one of the following datatypes:

disabled_for_team Void This user's Dropbox Business team doesn't allow file requests.

/create

VERSION 1 ▾

DESCRIPTION Creates a file request for this user.

URL STRUCTURE https://api.dropboxapi.com/2/file_requests/create

AUTHENTICATION [User Authentication](#)

ENDPOINT FORMAT [RPC](#)

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/file_requests/create \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"title\": \"Homework submission\", \"destination\": \"/File Requests/Homework\", \"deadline\": {\"deadline\": \"2020-10-12T17:00:00Z\", \"allow_late_uploads\": \"seven_days\"}}
```

PARAMETERS

```
{ "title": "Homework submission", "destination": "/File Requests/Homework", "deadline": { "deadline": "2020-10-12T17:00:00Z", "allow_late_uploads": "seven_days" }, }
```



createFileRequestArgs

Arguments for `create`.

title `String(min_length=1)` The title of the file request. Must not be empty.

destination `String(pattern="/(.|[r\nJ])*")` The path of the folder in the Dropbox where uploaded files will be sent. For apps with the app folder permission, this will be relative to the app folder.

deadline `FileRequestDeadline?` The deadline for the file request. Deadlines can only be set by Professional and Business accounts. This field is optional.

open `Boolean` Whether or not the file request should be open. If the file request is closed, it will not accept any file submissions, but it can be opened later. The default for this field is True.

RETURNS

```
{  
    "id": "oaCAVmEyrqYnkZX9955Y",  
    "url": "https://www.dropbox.com/request/oaCAVmEyrqYnkZX9955Y",  
    "title": "Homework submission",  
    "created": "2015-10-05T17:00:00Z",  
    "is_open": true,  
    "file_count": 3,  
    "destination": "/File Requests/Homework",  
    "deadline": {  
        "deadline": "2020-10-12T17:00:00Z",  
        "allow_late_uploads": {  
            ".tag": "seven_days"  
        }  
    }  
}
```

Example: with_deadline

```
{  
    "id": "BAJ7IrRGicQKGToykQdB",  
    "url": "https://www.dropbox.com/request/BAJ7IrRGjcQKGToykQdB",  
    "title": "Photo contest submission",  
    "created": "2015-11-02T04:00:00Z",  
    "is_open": true,  
    "file_count": 105,  
    "destination": "/Photo contest entries",  
    "deadline": {  
        "deadline": "2020-10-12T17:00:00Z"  
    }  
}
```

Example: with_no_deadline

```
{  
    "id": "rxwMPvK3ATTa0Vx0Ju5T",  
    "url": "https://www.dropbox.com/request/rxwMPvK3ATTa0Vx0Ju5T",  
    "title": "Wedding photo submission",  
    "created": "2015-12-15T13:02:00Z",  
    "is_open": true,  
    "file_count": 37,  
    "destination": "/Wedding photos"  
}
```

FileRequest

A [file request](#) for receiving files into the user's Dropbox account.

id `String(min_length=1, pattern="[-_0-9a-zA-Z]+")` The ID of the file request.

url `String(min_length=1)` The URL of the file request.

title `String(min_length=1)` The title of the file request.

created `Timestamp(format="%Y-%m-%dT%H:%M:%SZ")` When this file request was created.

is_open `Boolean` Whether or not the file request is open. If the file request is closed, it will not accept any more file submissions.

file_count `Int64` The number of files this file request has received.

destination `String(pattern="/(.|[r\nJ])*")?` The path of the folder in the Dropbox where uploaded files will be sent. This can be None if the destination was removed. For apps with the app folder permission, this will be relative to the app folder. This field is optional.

deadline `FileRequestDeadline?` The deadline for this file request. Only set if the request has a deadline. This field is optional.

ERRORS

Example: disabled_for_team

```
{  
    "error_summary": "disabled_for_team/...",  
    "error": {  
        ".tag": "disabled_for_team"  
    }  
}
```

Example: other



```
        ".tag": "other"
    }
```

Example: `not_found`

```
{
    "error_summary": "not_found/...",
    "error": {
        ".tag": "not_found"
    }
}
```

Example: `not_a_folder`

```
{
    "error_summary": "not_a_folder/...",
    "error": {
        ".tag": "not_a_folder"
    }
}
```

Example: `app_lacks_access`

```
{
    "error_summary": "app_lacks_access/...",
    "error": {
        ".tag": "app_lacks_access"
    }
}
```

Example: `no_permission`

```
{
    "error_summary": "no_permission/...",
    "error": {
        ".tag": "no_permission"
    }
}
```

Example: `email_unverified`

```
{
    "error_summary": "email_unverified/...",
    "error": {
        ".tag": "email_unverified"
    }
}
```

Example: `validation_error`

```
{
    "error_summary": "validation_error/...",
    "error": {
        ".tag": "validation_error"
    }
}
```

Example: `invalid_location`

```
{
    "error_summary": "invalid_location/...",
    "error": {
        ".tag": "invalid_location"
    }
}
```

Example: `rate_limit`

```
{
    "error_summary": "rate_limit/...",
    "error": {
        ".tag": "rate_limit"
    }
}
```

CreateFileRequestError (*union*)

There was an error creating the file request. The value will be one of the following datatypes:

disabled_for_team *Void* This user's Dropbox Business team doesn't allow file requests.

not_found *Void* This file request ID was not found.

not_a_folder *Void* The specified path is not a folder.

app_lacks_access *Void* This file request is not accessible to this app. Apps with the app folder permission can only access file requests in their app folder.

no_permission *Void* This user doesn't have permission to access or modify this file request.



validation_error *Void* There was an error validating the request.

For example, the title was invalid, or there were disallowed characters in the destination path.

invalid_location *Void* File requests are not available on the specified folder.

rate_limit *Void* The user has reached the rate limit for creating file requests. The limit is currently 4000 file requests total.

/delete

VERSION

1

DESCRIPTION Delete a batch of closed file requests.

URL STRUCTURE

https://api.dropboxapi.com/2/file_requests/delete

AUTHENTICATION

User Authentication

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/file_requests/delete \
    --header "Authorization: Bearer <get access token>" \
    --header "Content-Type: application/json" \
    --data "{\"ids\": [\"oaCAVmEyrqYnkZX9955Y\", \"BaZmehYoXMPtaRmfTbSG\"]}"
```

PARAMETERS

```
{
  "ids": [
    "oaCAVmEyrqYnkZX9955Y",
    "BaZmehYoXMPtaRmfTbSG"
  ]
}
```

DeleteFileRequestArgs

Arguments for [delete](#).

ids *List of (String(min_length=1, pattern="[-_0-9a-zA-Z]+"))* List IDs of the file requests to delete.

RETURNS

```
{
  "file_requests": [
    {
      "id": "oaCAVmEyrqYnkZX9955Y",
      "url": "https://www.dropbox.com/request/oaCAVmEyrqYnkZX9955Y",
      "title": "Homework submission",
      "created": "2015-10-05T17:00:00Z",
      "is_open": true,
      "file_count": 3,
      "destination": "/File Requests/Homework",
      "deadline": {
        "deadline": "2020-10-12T17:00:00Z",
        "allow_late_uploads": {
          ".tag": "seven_days"
        }
      }
    },
    {
      "id": "BAJ7IrRGicQKGToykQdB",
      "url": "https://www.dropbox.com/request/BAJ7IrRGj cQKGToykQdB",
      "title": "Photo contest submission",
      "created": "2015-11-02T04:00:00Z",
      "is_open": true,
      "file_count": 105,
      "destination": "/Photo contest entries",
      "deadline": {
        "deadline": "2020-10-12T17:00:00Z"
      }
    },
    {
      "id": "rxwMPvK3ATTa0Vx0Ju5T",
      "url": "https://www.dropbox.com/request/rxwMPvK3ATTa0Vx0Ju5T",
      "title": "Wedding photo submission",
      "created": "2015-12-15T13:02:00Z",
      "is_open": true,
      "file_count": 37,
      "destination": "/Wedding photos"
    }
  ]
}
```

DeleteFileRequestsResult

Result for [delete](#).

file_requests *List of (FileRequest)* The file requests deleted by the request.

ERRORS



≡

```
{
  "error_summary": "disabled_for_team/...",
  "error": {
    ".tag": "disabled_for_team"
  }
}
```

Example: other

```
{
  "error_summary": "other/...",
  "error": {
    ".tag": "other"
  }
}
```

Example: not_found

```
{
  "error_summary": "not_found/...",
  "error": {
    ".tag": "not_found"
  }
}
```

Example: not_a_folder

```
{
  "error_summary": "not_a_folder/...",
  "error": {
    ".tag": "not_a_folder"
  }
}
```

Example: app_lacks_access

```
{
  "error_summary": "app_lacks_access/...",
  "error": {
    ".tag": "app_lacks_access"
  }
}
```

Example: no_permission

```
{
  "error_summary": "no_permission/...",
  "error": {
    ".tag": "no_permission"
  }
}
```

Example: email_unverified

```
{
  "error_summary": "email_unverified/...",
  "error": {
    ".tag": "email_unverified"
  }
}
```

Example: validation_error

```
{
  "error_summary": "validation_error/...",
  "error": {
    ".tag": "validation_error"
  }
}
```

Example: file_request_open

```
{
  "error_summary": "file_request_open/...",
  "error": {
    ".tag": "file_request_open"
  }
}
```

DeleteFileRequestError (*union*)

There was an error deleting these file requests. The value will be one of the following datatypes:

disabled_for_team *Void* This user's Dropbox Business team doesn't allow file requests.

not_found *Void* This file request ID was not found.

not_a_folder *Void* The specified path is not a folder.

app_lacks_access *Void* This file request is not accessible to this app. Apps with the app folder permission can only access file requests in their app folder.



is or

1.

File requests are only available on accounts with a verified email address. Users can verify their email address [here](#).

validation_error Void There was an error validating the request. For example, the title was invalid, or there were disallowed characters in the destination path.

file_request_open Void One or more file requests currently open.

/delete_all_closed

VERSION

1

DESCRIPTION Delete all closed file requests owned by this user.

URL STRUCTURE

https://api.dropboxapi.com/2/file_requests/delete_all_closed

AUTHENTICATION [User Authentication](#)

ENDPOINT
FORMAT

EXAMPLE Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/file_requests/delete_all_closed \
--header "Authorization: Bearer <get access token>"
```

PARAMETERS No parameters.

RETURNS

```
{
  "file_requests": [
    {
      "id": "oaCAVmEyrqYnkZX9955Y",
      "url": "https://www.dropbox.com/request/oaCAVmEyrqYnkZX9955Y",
      "title": "Homework submission",
      "created": "2015-10-05T17:00:00Z",
      "is_open": true,
      "file_count": 3,
      "destination": "/File Requests/Homework",
      "deadline": {
        "deadline": "2020-10-12T17:00:00Z",
        "allow_late_uploads": {
          ".tag": "seven_days"
        }
      }
    },
    {
      "id": "BAJ7IrRGicQKGToykQdB",
      "url": "https://www.dropbox.com/request/BAJ7IrRGjcQKGToykQdB",
      "title": "Photo contest submission",
      "created": "2015-11-02T04:00:00Z",
      "is_open": true,
      "file_count": 105,
      "destination": "/Photo contest entries",
      "deadline": {
        "deadline": "2020-10-12T17:00:00Z"
      }
    },
    {
      "id": "rxwMPvK3ATTa0Vx0Ju5T",
      "url": "https://www.dropbox.com/request/rxwMPvK3ATTa0Vx0Ju5T",
      "title": "Wedding photo submission",
      "created": "2015-12-15T13:02:00Z",
      "is_open": true,
      "file_count": 37,
      "destination": "/Wedding photos"
    }
  ]
}
```

DeleteAllClosedFileRequestsResult

Result for [delete_all_closed](#).

file_requests List of (*FileRequest*) The file requests deleted for this user.

ERRORS Example: disabled_for_team

```
{
  "error_summary": "disabled_for_team/...",
  "error": {
    ".tag": "disabled_for_team"
  }
}
```

Example: other

```
{
  "error_summary": "other/...",
```



≡

Example: not_found

```
{  
    "error_summary": "not_found/...",  
    "error": {  
        ".tag": "not_found"  
    }  
}
```

Example: not_a_folder

```
{  
    "error_summary": "not_a_folder/...",  
    "error": {  
        ".tag": "not_a_folder"  
    }  
}
```

Example: app_lacks_access

```
{  
    "error_summary": "app_lacks_access/...",  
    "error": {  
        ".tag": "app_lacks_access"  
    }  
}
```

Example: no_permission

```
{  
    "error_summary": "no_permission/...",  
    "error": {  
        ".tag": "no_permission"  
    }  
}
```

Example: email_unverified

```
{  
    "error_summary": "email_unverified/...",  
    "error": {  
        ".tag": "email_unverified"  
    }  
}
```

Example: validation_error

```
{  
    "error_summary": "validation_error/...",  
    "error": {  
        ".tag": "validation_error"  
    }  
}
```

DeleteAllClosedFileRequestsError (*union*)

There was an error deleting all closed file requests. The value will be one of the following datatypes:

disabled_for_team *Void* This user's Dropbox Business team doesn't allow file requests.

not_found *Void* This file request ID was not found.

not_a_folder *Void* The specified path is not a folder.

app_lacks_access *Void* This file request is not accessible to this app. Apps with the app folder permission can only access file requests in their app folder.

no_permission *Void* This user doesn't have permission to access or modify this file request.

email_unverified *Void* This user's email address is not verified. File requests are only available on accounts with a verified email address. Users can verify their email address [here](#).

validation_error *Void* There was an error validating the request. For example, the title was invalid, or there were disallowed characters in the destination path.

/get

VERSION

1

DESCRIPTION Returns the specified file request.

URL STRUCTURE

https://api.dropboxapi.com/2/file_requests/get



FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/file_requests/get \
  -header "Authorization: Bearer <get access token>" \
  -header "Content-Type: application/json" \
  --data "{\"id\": \"oaCAVmEyrqYnkZX9955Y\"}"
```

PARAMETERS

```
{  
  "id": "oaCAVmEyrqYnkZX9955Y"  
}
```

GetFileRequestArgs

Arguments for [get](#).

id *String(min_length=1, pattern="[-_0-9a-zA-Z]+")* The ID of the file request to retrieve.

RETURNS

```
{  
  "id": "oaCAVmEyrqYnkZX9955Y",  
  "url": "https://www.dropbox.com/request/oaCAVmEyrqYnkZX9955Y",  
  "title": "Homework submission",  
  "created": "2015-10-05T17:00:00Z",  
  "is_open": true,  
  "file_count": 3,  
  "destination": "/File Requests/Homework",  
  "deadline": {  
    "deadline": "2020-10-12T17:00:00Z",  
    "allow_late_uploads": {  
      ".tag": "seven_days"  
    }  
  }  
}
```

Example: with_deadline

```
{  
  "id": "BAJ7IrRGicQKGToykQdB",  
  "url": "https://www.dropbox.com/request/BAJ7IrRGjcQKGToykQdB",  
  "title": "Photo contest submission",  
  "created": "2015-11-02T04:00:00Z",  
  "is_open": true,  
  "file_count": 105,  
  "destination": "/Photo contest entries",  
  "deadline": {  
    "deadline": "2020-10-12T17:00:00Z"  
  }  
}
```

Example: with_no_deadline

```
{  
  "id": "rxwMPvK3ATTa0Vx0Ju5T",  
  "url": "https://www.dropbox.com/request/rxwMPvK3ATTa0Vx0Ju5T",  
  "title": "Wedding photo submission",  
  "created": "2015-12-15T13:02:00Z",  
  "is_open": true,  
  "file_count": 37,  
  "destination": "/Wedding photos"  
}
```

FileRequest

A [file request](#) for receiving files into the user's Dropbox account.

id *String(min_length=1, pattern="[-_0-9a-zA-Z]+")* The ID of the file request.

url *String(min_length=1)* The URL of the file request.

title *String(min_length=1)* The title of the file request.

created *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")* When this file request was created.

is_open *Boolean* Whether or not the file request is open. If the file request is closed, it will not accept any more file submissions.

file_count *Int64* The number of files this file request has received.

destination *String(pattern="/(.|[\r\n])*)?"* The path of the folder in the Dropbox where uploaded files will be sent. This can be None if the destination was removed. For apps with the app folder permission, this will be relative to the app folder. This field is optional.

deadline *FileRequestDeadline?* The deadline for this file request. Only set if the request has a deadline. This field is optional.

ERRORS

Example: disabled_for_team

```
{  
  "error_summary": "disabled_for_team...",  
  "error": {  
    ".tag": "disabled_for_team"
```



≡



Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: not_found

```
{  
    "error_summary": "not_found/...",  
    "error": {  
        ".tag": "not_found"  
    }  
}
```

Example: not_a_folder

```
{  
    "error_summary": "not_a_folder/...",  
    "error": {  
        ".tag": "not_a_folder"  
    }  
}
```

Example: app_lacks_access

```
{  
    "error_summary": "app_lacks_access/...",  
    "error": {  
        ".tag": "app_lacks_access"  
    }  
}
```

Example: no_permission

```
{  
    "error_summary": "no_permission/...",  
    "error": {  
        ".tag": "no_permission"  
    }  
}
```

Example: email_unverified

```
{  
    "error_summary": "email_unverified/...",  
    "error": {  
        ".tag": "email_unverified"  
    }  
}
```

Example: validation_error

```
{  
    "error_summary": "validation_error/...",  
    "error": {  
        ".tag": "validation_error"  
    }  
}
```

GetFileRequestError (*union*)

There was an error retrieving the specified file request. The value will be one of the following datatypes:

disabled_for_team *Void* This user's Dropbox Business team doesn't allow file requests.

not_found *Void* This file request ID was not found.

not_a_folder *Void* The specified path is not a folder.

app_lacks_access *Void* This file request is not accessible to this app. Apps with the app folder permission can only access file requests in their app folder.

no_permission *Void* This user doesn't have permission to access or modify this file request.

email_unverified *Void* This user's email address is not verified. File requests are only available on accounts with a verified email address. Users can verify their email address [here](#).

validation_error *Void* There was an error validating the request. For example, the title was invalid, or there were disallowed characters in the destination path.



≡

on,

this will only return file requests with destinations in the app folder.

URL STRUCTURE

```
https://api.dropboxapi.com/2/file_requests/list_v2
```

AUTHENTICATION

User Authentication

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/file_requests/list_v2 \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"limit\": 1000}"
```

PARAMETERS

```
{  
    "limit": 1000  
}
```

ListFileRequestsArg

Arguments for [list:2](#).

limit *UInt64* The maximum number of file requests that should be returned per request. The default for this field is 1000.

RETURNS

```
{  
    "file_requests": [  
        {  
            "id": "oaCAVmEyrqYnkZX9955Y",  
            "url": "https://www.dropbox.com/request/oaCAVmEyrqYnkZX9955Y",  
            "title": "Homework submission",  
            "created": "2015-10-05T17:00:00Z",  
            "is_open": true,  
            "file_count": 3,  
            "destination": "/File Requests/Homework",  
            "deadline": {  
                "deadline": "2020-10-12T17:00:00Z",  
                "allow_late_uploads": {  
                    ".tag": "seven_days"  
                }  
            },  
            {  
                "id": "BAJ7IrRGicQKGToykQdB",  
                "url": "https://www.dropbox.com/request/BAJ7IrRGjcQKGToykQdB",  
                "title": "Photo contest submission",  
                "created": "2015-11-02T04:00:00Z",  
                "is_open": true,  
                "file_count": 105,  
                "destination": "/Photo contest entries",  
                "deadline": {  
                    "deadline": "2020-10-12T17:00:00Z"  
                }  
            },  
            {  
                "id": "rxwMPvK3ATTa0Vx0Ju5T",  
                "url": "https://www.dropbox.com/request/rxwMPvK3ATTa0Vx0Ju5T",  
                "title": "Wedding photo submission",  
                "created": "2015-12-15T13:02:00Z",  
                "is_open": true,  
                "file_count": 37,  
                "destination": "/Wedding photos"  
            },  
        ],  
        "cursor": "ZtkX9_EHj3x7PMkVuFIhwKYXEpwplwyxp9vMKomUhllil9q7eWiAu",  
        "has_more": true  
    }  
}
```

ListFileRequestsV2Result

Result for [list:2](#) and [list/continue](#).

file_requests *List of (FileRequest)* The file requests owned by this user. Apps with the app folder permission will only see file requests in their app folder.

cursor *String* Pass the cursor into [list/continue](#) to obtain additional file requests.

has_more *Boolean* Is true if there are additional file requests that have not been returned yet. An additional call to `:route:list/continue`` can retrieve them.

ERRORS

Example: disabled_for_team

```
{  
    "error_summary": "disabled_for_team/...",  
    "error": {  
        ".tag": "disabled_for_team"  
    }  
}
```



```
"error_summary": "other/...",
"error": {
    ".tag": "other"
}
```

ListFileRequestsError (*union*)

There was an error retrieving the file requests. The value will be one of the following datatypes:

disabled_for_team *Void* This user's Dropbox Business team doesn't allow file requests.

/list/continue

VERSION

1

PREVIEW - may change or disappear without notice

DESCRIPTION Once a cursor has been retrieved from [list:2](#), use this to paginate through all file requests. The cursor must come from a previous call to [list:2](#) or [list/continue](#).

URL STRUCTURE

https://api.dropboxapi.com/2/file_requests/list/continue

AUTHENTICATION

[User Authentication](#)

ENDPOINT
FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/file_requests/list/continue \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"cursor\": \"ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu\"}"
```

PARAMETERS

```
{\n    \"cursor\": \"ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu\"\n}
```

ListFileRequestsContinueArg

cursor *String* The cursor returned by the previous API call specified in the endpoint description.

RETURNS

```
{\n    \"file_requests\": [\n        {\n            \"id\": \"oaCAVmEyrqYnkZX9955Y\", \n            \"url\": \"https://www.dropbox.com/request/oaCAVmEyrqYnkZX9955Y\", \n            \"title\": \"Homework submission\", \n            \"created\": \"2015-10-05T17:00:00Z\", \n            \"is_open\": true, \n            \"file_count\": 3, \n            \"destination\": \"/File Requests/Homework\", \n            \"deadline\": {\n                \"deadline\": \"2020-10-12T17:00:00Z\", \n                \"allow_late_uploads\": {\n                    \".tag\": \"seven_days\"\n                }\n            }\n        },\n        {\n            \"id\": \"BAJ7IrRGicQKGToykQdB\", \n            \"url\": \"https://www.dropbox.com/request/BAJ7IrRGjcQKGToykQdB\", \n            \"title\": \"Photo contest submission\", \n            \"created\": \"2015-11-02T04:00:00Z\", \n            \"is_open\": true, \n            \"file_count\": 105, \n            \"destination\": \"/Photo contest entries\", \n            \"deadline\": {\n                \"deadline\": \"2020-10-12T17:00:00Z\"\n            }\n        },\n        {\n            \"id\": \"rxwMPvK3ATTa0Vx0Ju5T\", \n            \"url\": \"https://www.dropbox.com/request/rxwMPvK3ATTa0Vx0Ju5T\", \n            \"title\": \"Wedding photo submission\", \n            \"created\": \"2015-12-15T13:02:00Z\", \n            \"is_open\": true, \n            \"file_count\": 37, \n            \"destination\": \"/Wedding photos\"\n        }\n    ],\n    \"cursor\": \"ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu\", \n    \"has_more\": true\n}
```

ListFileRequestsV2Result

Result for [list:2](#) and [list/continue](#).



cursor *String* Pass the cursor into `list/continue` to obtain additional file requests.

has_more *Boolean* Is true if there are additional file requests that have not been returned yet. An additional call to `:route:list/continue`` can retrieve them.

ERRORS Example: `disabled_for_team`

```
{  
    "error_summary": "disabled_for_team/...",  
    "error": {  
        ".tag": "disabled_for_team"  
    }  
}
```

Example: `other`

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: `invalid_cursor`

```
{  
    "error_summary": "invalid_cursor/...",  
    "error": {  
        ".tag": "invalid_cursor"  
    }  
}
```

`ListFileRequestsContinueError (union)`

There was an error retrieving the file requests. The value will be one of the following datatypes:

disabled_for_team *Void* This user's Dropbox Business team doesn't allow file requests.

invalid_cursor *Void* The cursor is invalid.

/update

VERSION

1 ▾

DESCRIPTION Update a file request.

URL STRUCTURE

https://api.dropboxapi.com/2/file_requests/update

AUTHENTICATION [User Authentication](#)

ENDPOINT
FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/file_requests/update \  
--header "Authorization: Bearer <get access token>" \  
--header "Content-Type: application/json" \  
--data "{\"id\": \"oaCAVmEyrqYnkZX9955Y\", \"title\": \"Homework submission\", \"destination\": \"/File Requests/Homework\", \"deadline\": {\".tag\": \"update\", \"deadline\": \"2020-10-12T17:00:00Z\", \"allow_late_uploads\": \"seven_days\"}, \"open\": true}
```

PARAMETERS

```
{  
    "id": "oaCAVmEyrqYnkZX9955Y",  
    "title": "Homework submission",  
    "destination": "/File Requests/Homework",  
    "deadline": {  
        ".tag": "update",  
        "deadline": "2020-10-12T17:00:00Z",  
        "allow_late_uploads": "seven_days"  
    },  
    "open": true  
}
```

`UpdateFileRequestArgs`

Arguments for `update`.

id *String(min_length=1, pattern="[-_0-9a-zA-Z]+")* The ID of the file request to update.

title *String(min_length=1)?* The new title of the file request. Must not be empty. This field is optional.

destination *String(pattern="/(.|[\r\n])*)?")* The new path of the folder in the Dropbox where uploaded files will be sent. For apps with the app folder permission, this will be relative to the app folder. This field is optional.



open Boolean? Whether to set this file request as open or closed.
This field is optional.

RETURNS

```
{  
    "id": "oaCAVmEYrqYnkZX9955Y",  
    "url": "https://www.dropbox.com/request/oaCAVmEYrqYnkZX9955Y",  
    "title": "Homework submission",  
    "created": "2015-10-05T17:00:00Z",  
    "is_open": true,  
    "file_count": 3,  
    "destination": "/File Requests/Homework",  
    "deadline": {  
        "deadline": "2020-10-12T17:00:00Z",  
        "allow_late_uploads": {  
            ".tag": "seven_days"  
        }  
    }  
}
```

Example: with_deadline

```
{  
    "id": "BAJ7IrRGicQKGToykQdB",  
    "url": "https://www.dropbox.com/request/BAJ7IrRGjcQKGToykQdB",  
    "title": "Photo contest submission",  
    "created": "2015-11-02T04:00:00Z",  
    "is_open": true,  
    "file_count": 105,  
    "destination": "/Photo contest entries",  
    "deadline": {  
        "deadline": "2020-10-12T17:00:00Z"  
    }  
}
```

Example: with_no_deadline

```
{  
    "id": "rxwMPvK3ATTa0Vx0Ju5T",  
    "url": "https://www.dropbox.com/request/rxwMPvK3ATTa0Vx0Ju5T",  
    "title": "Wedding photo submission",  
    "created": "2015-12-15T13:02:00Z",  
    "is_open": true,  
    "file_count": 37,  
    "destination": "/Wedding photos"  
}
```

FileRequest

A [file request](#) for receiving files into the user's Dropbox account.
id String(min_length=1, pattern="[-_0-9a-zA-Z]+") The ID of the file request.

url String(min_length=1) The URL of the file request.

title String(min_length=1) The title of the file request.

created Timestamp(format="%Y-%m-%dT%H:%M:%SZ") When this file request was created.

is_open Boolean Whether or not the file request is open. If the file request is closed, it will not accept any more file submissions.

file_count Int64 The number of files this file request has received.

destination String(pattern="/(.|[\\r\\n])*)?") The path of the folder in the Dropbox where uploaded files will be sent. This can be None if the destination was removed. For apps with the app folder permission, this will be relative to the app folder. This field is optional.

deadline FileRequestDeadline? The deadline for this file request. Only set if the request has a deadline. This field is optional.

ERRORS

Example: disabled_for_team

```
{  
    "error_summary": "disabled_for_team/...",  
    "error": {  
        ".tag": "disabled_for_team"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: not_found

```
{  
    "error_summary": "not_found/...",  
}
```



≡

Example: not_a_folder

```
{  
    "error_summary": "not_a_folder/...",  
    "error": {  
        ".tag": "not_a_folder"  
    }  
}
```

Example: app_lacks_access

```
{  
    "error_summary": "app_lacks_access/...",  
    "error": {  
        ".tag": "app_lacks_access"  
    }  
}
```

Example: no_permission

```
{  
    "error_summary": "no_permission/...",  
    "error": {  
        ".tag": "no_permission"  
    }  
}
```

Example: email_unverified

```
{  
    "error_summary": "email_unverified/...",  
    "error": {  
        ".tag": "email_unverified"  
    }  
}
```

Example: validation_error

```
{  
    "error_summary": "validation_error/...",  
    "error": {  
        ".tag": "validation_error"  
    }  
}
```

UpdateFileRequestError (*union*)

There is an error updating the file request. The value will be one of the following datatypes:

disabled_for_team *Void* This user's Dropbox Business team doesn't allow file requests.

not_found *Void* This file request ID was not found.

not_a_folder *Void* The specified path is not a folder.

app_lacks_access *Void* This file request is not accessible to this app. Apps with the app folder permission can only access file requests in their app folder.

no_permission *Void* This user doesn't have permission to access or modify this file request.

email_unverified *Void* This user's email address is not verified. File requests are only available on accounts with a verified email address. Users can verify their email address [here](#).

validation_error *Void* There was an error validating the request. For example, the title was invalid, or there were disallowed characters in the destination path.

files

This namespace contains endpoints and data types for basic file operations.

/copy

VERSION

2

DESCRIPTION

Copy a file or folder to a different location in the user's Dropbox. If the source path is a folder all its contents will be copied.

URL STRUCTURE

https://api.dropboxapi.com/2/files/copy_v2

AUTHENTICATION

[User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)



```
curl -X POST https://api.dropboxapi.com/2/files/copy_v2 \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"from_path\": \"/Homework/math\", \"to_path\": \"/Homework/algebra\", \"allow_shared_folder\": false, \"allow_ownership_transfer\": false}
```

PARAMETERS

```
{  
    "from_path": "/Homework/math",  
    "to_path": "/Homework/algebra",  
    "allow_shared_folder": false,  
    "autorename": false,  
    "allow_ownership_transfer": false  
}
```

RelocationArg

from_path *String(pattern="(.|[\\r\\n])*|(ns:[0-9]+(.*?))|(id:.*")")*
Path in the user's Dropbox to be copied or moved.

to_path *String(pattern="(.|[\\r\\n])*|(ns:[0-9]+(.*?))|(id:.*")")* Path in the user's Dropbox that is the destination.

allow_shared_folder *Boolean* If true, `copy` will copy contents in shared folder, otherwise `RelocationError.cant_copy_shared_folder` will be returned if `from_path` contains shared folder. This field is always true for `move`. The default for this field is False.

autorename *Boolean* If there's a conflict, have the Dropbox server try to autorename the file to avoid the conflict. The default for this field is False.

allow_ownership_transfer *Boolean* Allow moves by owner even if it would result in an ownership transfer for the content being moved. This does not apply to copies. The default for this field is False.

RETURNS

```
{  
    "metadata": {  
        ".tag": "file",  
        "name": "Prime_Numbers.txt",  
        "id": "id:a4ayc_80_0EAAAAAAAAXw",  
        "client_modified": "2015-05-12T15:50:38Z",  
        "server_modified": "2015-05-12T15:50:38Z",  
        "rev": "a1c10ce0dd78",  
        "size": 7212,  
        "path_lower": "/homework/math/prime_numbers.txt",  
        "path_display": "/Homework/math/Prime_Numbers.txt",  
        "sharing_info": {  
            "read_only": true,  
            "parent_shared_folder_id": "84528192421",  
            "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"  
        },  
        "is_downloadable": true,  
        "property_groups": [  
            {  
                "template_id": "ptid:la5n2i6d30YEAFFFFFFAYa",  
                "fields": [  
                    {  
                        "name": "Security Policy",  
                        "value": "Confidential"  
                    }  
                ]  
            }  
        ],  
        "has_explicit_shared_members": false,  
        "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",  
        "file_lock_info": {  
            "is_lockholder": true,  
            "lockholder_name": "Imaginary User",  
            "created": "2015-05-12T15:50:38Z"  
        }  
    }  
}
```

RelocationResult

metadata *Metadata* Metadata of the relocated object.

ERRORS

Example: `cant_copy_shared_folder`

```
{  
    "error_summary": "cant_copy_shared_folder/...",  
    "error": {  
        ".tag": "cant_copy_shared_folder"  
    }  
}
```

Example: `cant_nest_shared_folder`

```
{  
    "error_summary": "cant_nest_shared_folder/...",  
    "error": {  
        ".tag": "cant_nest_shared_folder"  
    }  
}
```

Example: `cant_move_folder_into_itself`



≡

```
        ".tag": "cant_move_folder_into_itself"
    }
```

Example: too_many_files

```
{
    "error_summary": "too_many_files/...",
    "error": {
        ".tag": "too_many_files"
    }
}
```

Example: duplicated_or_nested_paths

```
{
    "error_summary": "duplicated_or_nested_paths/...",
    "error": {
        ".tag": "duplicated_or_nested_paths"
    }
}
```

Example: cant_transfer_ownership

```
{
    "error_summary": "cant_transfer_ownership/...",
    "error": {
        ".tag": "cant_transfer_ownership"
    }
}
```

Example: insufficient_quota

```
{
    "error_summary": "insufficient_quota/...",
    "error": {
        ".tag": "insufficient_quota"
    }
}
```

Example: internal_error

```
{
    "error_summary": "internal_error/...",
    "error": {
        ".tag": "internal_error"
    }
}
```

Example: cant_move_shared_folder

```
{
    "error_summary": "cant_move_shared_folder/...",
    "error": {
        ".tag": "cant_move_shared_folder"
    }
}
```

Example: other

```
{
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

RelocationError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

from_lookup *LookupError*
from_write *WriteError*
to *WriteError*
cant_copy_shared_folder *Void* Shared folders can't be copied.
cant_nest_shared_folder *Void* Your move operation would result in nested shared folders. This is not allowed.
cant_move_folder_into_itself *Void* You cannot move a folder into itself.
too_many_files *Void* The operation would involve more than 10,000 files and folders.
duplicated_or_nested_paths *Void* There are duplicated/nested paths among RelocationArg.from_path and RelocationArg.to_path.
cant_transfer_ownership *Void* Your move operation would result in an ownership transfer. You may reissue the request with the field RelocationArg.allow_ownership_transfer to true.



gh



Dropbox's end. You'll need to verify that the action you were taking succeeded, and if not, try again. This should happen very rarely.

cant_move_shared_folder *Void* Can't move the shared folder to the given destination.

/copy_batch

VERSION

2

DESCRIPTION Copy multiple files or folders to different locations at once in the user's Dropbox.
This route will replace [copy_batch:1](#). The main difference is this route will return status for each entry, while [copy_batch:1](#) raises failure if any entry fails.
This route will either finish synchronously, or return a job ID and do the async copy job in background. Please use [copy_batch/check:2](#) to check the job status.

URL STRUCTURE

https://api.dropboxapi.com/2/files/copy_batch_v2

AUTHENTICATION

User Authentication, Dropbox-API-Select-Admin (Team Admin)

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/files/copy_batch_v2 \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"entries\": [{\"from_path\": \"/Homework/math\", \"to_path\": \"/Homework/algebra\"}], \"autorename\": false}
```

PARAMETERS

```
{\n    \"entries\": [\n        {\n            \"from_path\": \"/Homework/math\",\n            \"to_path\": \"/Homework/algebra\"\n        }\n    ],\n    \"autorename\": false\n}
```

RelocationBatchArgBase

entries *List of (RelocationPath, min_items=1)* List of entries to be moved or copied. Each entry is RelocationPath.

autorename *Boolean* If there's a conflict with any file, have the Dropbox server try to autorename that file to avoid the conflict. The default for this field is False.

RETURNS

Example: complete

```
{\n    \".tag\": \"complete\",\n    \"entries\": [\n        {\n            \".tag\": \"success\",\n            \"success\": {\n                \".tag\": \"file\",\n                \"name\": \"Prime_Numbers.txt\",\n                \"id\": \"id:aayc_80_OEAAAAAAAAXw\",\n                \"client_modified\": \"2015-05-12T15:50:38Z\",\n                \"server_modified\": \"2015-05-12T15:50:38Z\",\n                \"rev\": \"a1c10ce0dd78\",\n                \"size\": 7212,\n                \"path_lower\": \"/homework/math/prime_numbers.txt\",\n                \"path_display\": \"/Homework/math/Prime_Numbers.txt\",\n                \"sharing_info\": {\n                    \"read_only\": true,\n                    \"parent_shared_folder_id\": \"84528192421\",\n                    \"modified_by\": \"dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc\"\n                },\n                \"is_downloadable\": true,\n                \"property_groups\": [\n                    {\n                        \"template_id\": \"ptid:la5n2i6d30YEAAAAAAAAYa\",\n                        \"fields\": [\n                            {\n                                \"name\": \"Security Policy\",\n                                \"value\": \"Confidential\"\n                            }\n                        ]\n                    }\n                ],\n                \"has_explicit_shared_members\": false,\n                \"content_hash\": \"e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855\",\n                \"file_lock_info\": {\n                    \"is_lockholder\": true,\n                    \"lockholder_name\": \"Imaginary User\",\n                    \"created\": \"2015-05-12T15:50:38Z\"\n                }\n            }\n        }\n    ]\n}
```



Example: `async_job_id`

```
{  
    ".tag": "async_job_id",  
    "async_job_id": "34g93hh34h04y384084"  
}
```

RelocationBatchV2Launch (*union*)

Result returned by `copy_batch:2` or `move_batch:2` that may either launch an asynchronous job or complete synchronously. The value will be one of the following datatypes:

async_job_id *String(min_length=1)* This response indicates that the processing is asynchronous. The string is an id that can be used to obtain the status of the asynchronous job.

complete *RelocationBatchV2Result*

ERRORS No errors.

/copy_batch/check

VERSION

2

DESCRIPTION Returns the status of an asynchronous job for [copy_batch:2](#). It returns list of results for each entry.

URL STRUCTURE https://api.dropboxapi.com/2/files/copy_batch/check_v2

AUTHENTICATION User Authentication, Dropbox-API-Select-Admin (Team Admin)

RPC

EXAMPLE

Get access token for: ProfesorRobo

```
curl -X POST https://api.dropboxapi.com/2/files/copy_batch/check_v2 \
    -header "Authorization: Bearer <get access token>" \
    -header "Content-Type: application/json" \
    --data "{\"async_job_id\": \"34g93hh34h04y384084\"}"
```

PARAMETERS

```
{  
    "async_job_id": "34g93hh34h04y384084"  
}
```

PollArg

Arguments for methods that poll the status of an asynchronous job. This datatype comes from an imported namespace originally defined in the `async` namespace.

async_job_id *String(min_length=1)* Id of the asynchronous job. This is the value of a response returned from the method that launched the job.

RETURNS

```
{  
    ".tag": "complete",  
    "entries": [  
        {  
            ".tag": "success",  
            "success": {  
                ".tag": "file",  
                "name": "Prime_Numbers.txt",  
                "id": "id:a4ayc_80_0EAAAAAAAAXw",  
                "client_modified": "2015-05-12T15:50:38Z",  
                "server_modified": "2015-05-12T15:50:38Z",  
                "rev": "alc10ce0dd78",  
                "size": 7212,  
                "path_lower": "/homework/math/prime_numbers.txt",  
                "path_display": "/Homework/math/Prime_Numbers.txt",  
                "sharing_info": {  
                    "read_only": true,  
                    "parent_shared_folder_id": "84528192421",  
                    "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngC"  
                },  
                "is_downloadable": true,  
                "property_groups": [  
                    {  
                        "template_id": "ptid:1a5n2i6d30YEAFFFFFFAYa",  
                        "fields": [  
                            {  
                                "name": "Security Policy",  
                                "value": "Confidential"  
                            }  
                        ]  
                    }  
                ]  
            }  
        }  
    ]  
}
```



!7ae41e4649b934ca495991b7852b855",

```
        "lockholder_name": "Imaginary User",
        "created": "2015-05-12T15:50:38Z"
    }
}
```

Example: `in_progress`

```
{\n    ".tag": "in_progress"\n}
```

`RelocationBatchV2JobStatus (union)`

Result returned by `copy_batch/check:2` or `move_batch/check:2` that may either be in progress or completed with result for each entry. The value will be one of the following datatypes:

in_progress *Void* The asynchronous job is still in progress.

complete *RelocationBatchV2Result* The copy or move batch job has finished.

ERRORS Example: `invalid_async_job_id`

```
{\n    "error_summary": "invalid_async_job_id/...",\n    "error": {\n        ".tag": "invalid_async_job_id"\n    }\n}
```

Example: `internal_error`

```
{\n    "error_summary": "internal_error/...",\n    "error": {\n        ".tag": "internal_error"\n    }\n}
```

Example: `other`

```
{\n    "error_summary": "other/...",\n    "error": {\n        ".tag": "other"\n    }\n}
```

`PollError (open union)`

Error returned by methods for polling the status of asynchronous job. This datatype comes from an imported namespace originally defined in the async namespace. The value will be one of the following datatypes. New values may be introduced as our API evolves.

invalid_async_job_id *Void* The job ID is invalid.

internal_error *Void* Something went wrong with the job on Dropbox's end. You'll need to verify that the action you were taking succeeded, and if not, try again. This should happen very rarely.

/copy_reference/get

VERSION

1

DESCRIPTION Get a copy reference to a file or folder. This reference string can be used to save that file or folder to another user's Dropbox by passing it to [copy_reference/save](#).

URL STRUCTURE

https://api.dropboxapi.com/2/files/copy_reference/get

AUTHENTICATION [User Authentication](#)

RPC

EXAMPLE Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/files/copy_reference/get \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"path\": \"/video.mp4\"}"
```

PARAMETERS



≡



GetCopyReferenceArg

path *String(pattern="/(.|[\\r\\n])*|id:.*|(rev:[0-9a-f]{9,})|(ns:[0-9]+(.*)?")* The path to the file or folder you want to get a copy reference to.

RETURNS

```
{  
    "metadata": {  
        ".tag": "file",  
        "name": "Prime_Numbers.txt",  
        "id": "id:a4ayc_80_0EAAAAAAAAXw",  
        "client_modified": "2015-05-12T15:50:38Z",  
        "server_modified": "2015-05-12T15:50:38Z",  
        "rev": "alc10ce0dd78",  
        "size": 7212,  
        "path_lower": "/homework/math/prime_numbers.txt",  
        "path_display": "/Homework/math/Prime_Numbers.txt",  
        "sharing_info": {  
            "read_only": true,  
            "parent_shared_folder_id": "84528192421",  
            "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"  
        },  
        "is_downloadable": true,  
        "property_groups": [  
            {  
                "template_id": "ptid:la5n2i6d30YEAAAAAAAAAYa",  
                "fields": [  
                    {  
                        "name": "Security Policy",  
                        "value": "Confidential"  
                    }  
                ]  
            }  
        ],  
        "has_explicit_shared_members": false,  
        "content_hash": "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",  
        "file_lock_info": {  
            "is_lockholder": true,  
            "lockholder_name": "Imaginary User",  
            "created": "2015-05-12T15:50:38Z"  
        }  
    },  
    "copy_reference": "z1X6ATl6aWtz0Gq0c3g5Ng",  
    "expires": "2045-05-12T15:50:38Z"  
}
```

GetCopyReferenceResult

metadata *Metadata* Metadata of the file or folder.

copy_reference *String* A copy reference to the file or folder.

expires *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")* The expiration date of the copy reference. This value is currently set to be far enough in the future so that expiration is effectively not an issue.

ERRORS

GetCopyReferenceError (open union)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path *LookupError*

/copy_reference/save

VERSION

1

DESCRIPTION

Save a copy reference returned by [copy_reference/get](#) to the user's Dropbox.

URL STRUCTURE

https://api.dropboxapi.com/2/files/copy_reference/save

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/files/copy_reference/save \  
    --header "Authorization: Bearer <get access token>" \  
    --header "Content-Type: application/json" \  
    --data "{\"copy_reference\": \"z1X6ATl6aWtz0Gq0c3g5Ng\", \"path\": \"/video.mp4\"}"
```

PARAMETERS

```
{  
    "copy_reference": "z1X6ATl6aWtz0Gq0c3g5Ng",  
    "path": "/video.mp4"  
}
```

SaveCopyReferenceArg



at is

the destination.

RETURNS

```
{  
    "metadata": {  
        ".tag": "file",  
        "name": "Prime_Numbers.txt",  
        "id": "id:a4ayc_80_0EAAAAAAAAXw",  
        "client_modified": "2015-05-12T15:50:38Z",  
        "server_modified": "2015-05-12T15:50:38Z",  
        "rev": "a1c10ce0dd78",  
        "size": 7212,  
        "path_lower": "/homework/math/prime_numbers.txt",  
        "path_display": "/Homework/math/Prime_Numbers.txt",  
        "sharing_info": {  
            "read_only": true,  
            "parent_shared_folder_id": "84528192421",  
            "modified_by": "dbid:AAH4f99T0ta0N1b-OurWxbNQ6ywGRopQngc"  
        },  
        "is_downloadable": true,  
        "property_groups": [  
            {  
                "template_id": "ptid:1a5n2i6d30YEAAAAAAAAYa",  
                "fields": [  
                    {  
                        "name": "Security Policy",  
                        "value": "Confidential"  
                    }  
                ]  
            }  
        ],  
        "has_explicit_shared_members": false,  
        "content_hash": "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",  
        "file_lock_info": {  
            "is_lockholder": true,  
            "lockholder_name": "Imaginary User",  
            "created": "2015-05-12T15:50:38Z"  
        }  
    }  
}
```

SaveCopyReferenceResult

metadata *Metadata* The metadata of the saved file or folder in the user's Dropbox.

ERRORS

Example: invalid_copy_reference

```
{  
    "error_summary": "invalid_copy_reference/...",  
    "error": {  
        ".tag": "invalid_copy_reference"  
    }  
}
```

Example: no_permission

```
{  
    "error_summary": "no_permission/...",  
    "error": {  
        ".tag": "no_permission"  
    }  
}
```

Example: not_found

```
{  
    "error_summary": "not_found/...",  
    "error": {  
        ".tag": "not_found"  
    }  
}
```

Example: too_many_files

```
{  
    "error_summary": "too_many_files/...",  
    "error": {  
        ".tag": "too_many_files"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

SaveCopyReferenceError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.



no_permission *Void* You don't have permission to save the given copy reference. Please make sure this app is same app which created the copy reference and the source user is still linked to the app.

not_found *Void* The file referenced by the copy reference cannot be found.

too_many_files *Void* The operation would involve more than 10,000 files and folders.

/create_folder

VERSION

2

DESCRIPTION Create a folder at a given path.

URL STRUCTURE

https://api.dropboxapi.com/2/files/create_folder_v2

AUTHENTICATION [User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT FORMAT

EXAMPLE Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/files/create_folder_v2 \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"path\": \"/Homework/math\", \"autorename\": false}"
```

PARAMETERS

```
{
  "path": "/Homework/math",
  "autorename": false
}
```

CreateFolderArg

path *String(pattern="/(.|[r\nJ])*|(ns:[0-9]+(.*?))")* Path in the user's Dropbox to create.

autorename *Boolean* If there's a conflict, have the Dropbox server try to autorename the folder to avoid the conflict. The default for this field is False.

RETURNS

```
{
  "metadata": {
    "name": "math",
    "id": "id:aayc_80_0EAAAAAAAAXz",
    "path_lower": "/homework/math",
    "path_display": "/Homework/math",
    "sharing_info": {
      "read_only": false,
      "parent_shared_folder_id": "84528192421",
      "traverse_only": false,
      "no_access": false
    },
    "property_groups": [
      {
        "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",
        "fields": [
          {
            "name": "Security Policy",
            "value": "Confidential"
          }
        ]
      }
    ]
  }
}
```

CreateFolderResult

metadata *FolderMetadata* Metadata of the created folder.

ERRORS

CreateFolderError (union)

The value will be one of the following datatypes:

path *WriteError*

/create_folder_batch

VERSION

1

DESCRIPTION Create multiple folders at once.

This route is asynchronous for large batches, which returns a job ID immediately and runs the create folder batch asynchronously. Otherwise, creates the folders and returns the result synchronously for smaller inputs. You can force asynchronous behaviour by using the



URL STRUCTURE

```
https://api.dropboxapi.com/2/files/create_folder_batch
```

AUTHENTICATION

User Authentication, Dropbox-API-Select-Admin (Team Admin)

ENDPOINT
FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/create_folder_batch \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"paths\": [\"/Homework/math\"], \"autorename\": false, \"force_async\": false}"
```

PARAMETERS

```
{  
    "paths": [  
        "/Homework/math"  
    ],  
    "autorename": false,  
    "force_async": false  
}
```

CreateFolderBatchArg

paths List of (String(pattern="(/(.|[\\r\\n])*|(ns:[0-9]+(/.*?))") List of paths to be created in the user's Dropbox. Duplicate path arguments in the batch are considered only once.

autorename Boolean If there's a conflict, have the Dropbox server try to autorename the folder to avoid the conflict. The default for this field is False.

force_async Boolean Whether to force the create to happen asynchronously. The default for this field is False.

RETURNS

Example: complete

```
{  
    ".tag": "complete",  
    "entries": [  
        {  
            ".tag": "success",  
            "metadata": {  
                "name": "math",  
                "id": "id:a4ayc_80_OEAAAAAAAAXz",  
                "path_lower": "/homework/math",  
                "path_display": "/Homework/math",  
                "sharing_info": {  
                    "read_only": false,  
                    "parent_shared_folder_id": "84528192421",  
                    "traverse_only": false,  
                    "no_access": false  
                },  
                "property_groups": [  
                    {  
                        "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",  
                        "fields": [  
                            {  
                                "name": "Security Policy",  
                                "value": "Confidential"  
                            }  
                        ]  
                    }  
                ]  
            }  
        }  
    ]  
}
```

Example: async_job_id

```
{  
    ".tag": "async_job_id",  
    "async_job_id": "34g93hh34h04y384084"  
}
```

Example: other

```
{  
    ".tag": "other"  
}
```

CreateFolderBatchLaunch (*open union*)

Result returned by `create_folder_batch` that may either launch an asynchronous job or complete synchronously. The value will be one of the following datatypes. New values may be introduced as our API evolves.

async_job_id String(*min_length=1*) This response indicates that the processing is asynchronous. The string is an id that can be used to obtain the status of the asynchronous job.



/create_folder_batch/check

VERSION

1

DESCRIPTION Returns the status of an asynchronous job for [create_folder_batch](#). If success, it returns list of result for each entry.

URL STRUCTURE

https://api.dropboxapi.com/2/files/create_folder_batch/check

AUTHENTICATION [User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT FORMAT
RPC

EXAMPLE Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/files/create_folder_batch/check \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"async_job_id\": \"34g93hh34h04y384084\"}"
```

PARAMETERS

```
{  
    "async_job_id": "34g93hh34h04y384084"  
}
```

PollArg

Arguments for methods that poll the status of an asynchronous job. This datatype comes from an imported namespace originally defined in the `async` namespace.

async_job_id *String(min_length=1)* Id of the asynchronous job. This is the value of a response returned from the method that launched the job.

RETURNS

```
{  
    ".tag": "complete",  
    "entries": [  
        {  
            ".tag": "success",  
            "metadata": {  
                "name": "math",  
                "id": "id:a4ayc_80_0EAAAAAAAAXz",  
                "path_lower": "/homework/math",  
                "path_display": "/Homework/math",  
                "sharing_info": {  
                    "read_only": false,  
                    "parent_shared_folder_id": "84528192421",  
                    "traverse_only": false,  
                    "no_access": false  
                },  
                "property_groups": [  
                    {  
                        "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",  
                        "fields": [  
                            {  
                                "name": "Security Policy",  
                                "value": "Confidential"  
                            }  
                        ]  
                    }  
                ]  
            }  
        }  
    ]  
}
```

Example: in_progress

```
{  
    ".tag": "in_progress"  
}
```

Example: other

```
{  
    ".tag": "other"  
}
```

CreateFolderBatchJobStatus (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

in_progress *Void* The asynchronous job is still in progress.

complete *CreateFolderBatchResult* The batch create folder has finished.



```
{  
    "error_summary": "invalid_async_job_id/...",  
    "error": {  
        ".tag": "invalid_async_job_id"  
    }  
}
```

Example: internal_error

```
{  
    "error_summary": "internal_error/...",  
    "error": {  
        ".tag": "internal_error"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

PollError (*open union*)

Error returned by methods for polling the status of asynchronous job. This datatype comes from an imported namespace originally defined in the async namespace. The value will be one of the following datatypes. New values may be introduced as our API evolves.

invalid_async_job_id *Void* The job ID is invalid.

internal_error *Void* Something went wrong with the job on Dropbox's end. You'll need to verify that the action you were taking succeeded, and if not, try again. This should happen very rarely.

/delete

VERSION

2



DESCRIPTION Delete the file or folder at a given path.
If the path is a folder, all its contents will be deleted too.
A successful response indicates that the file or folder was deleted. The returned metadata will be the corresponding FileMetadata or FolderMetadata for the item at time of deletion, and not a DeletedMetadata object.

URL STRUCTURE

https://api.dropboxapi.com/2/files/delete_v2

AUTHENTICATION [User Authentication, Dropbox-API-Select-Admin \(Team Admin\)](#)

ENDPOINT
FORMAT

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/delete_v2 \  
    --header "Authorization: Bearer <get access token>" \  
    --header "Content-Type: application/json" \  
    --data "{\"path\": \"/Homework/math/Prime_Numbers.txt\"}"
```

PARAMETERS

```
{  
    "path": "/Homework/math/Prime_Numbers.txt"  
}
```

DeleteArg

path *String(pattern="(.|[r\nJ])*|(ns:[0-9]+(.*?)(id:.*))")* Path in the user's Dropbox to delete.

parent_rev *String(min_length=9, pattern="[0-9a-f]+")?* Perform delete if given "rev" matches the existing file's latest "rev". This field does not support deleting a folder. This field is optional.

RETURNS

```
{  
    "metadata": {  
        ".tag": "file",  
        "name": "Prime_Numbers.txt",  
        "id": "id:a4ayc_80_0EAAAAAAAAXw",  
        "client_modified": "2015-05-12T15:50:38Z",  
        "server_modified": "2015-05-12T15:50:38Z",  
        "rev": "alc10ce0dd78",  
        "size": 7212,  
        "path_lower": "/homework/math/prime_numbers.txt",  
    }  
}
```



≡

```
        "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"
    },
    "is_downloadable": true,
    "property_groups": [
        {
            "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",
            "fields": [
                {
                    "name": "Security Policy",
                    "value": "Confidential"
                }
            ]
        }
    ],
    "has_explicit_shared_members": false,
    "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",
    "file_lock_info": {
        "is_lockholder": true,
        "lockholder_name": "Imaginary User",
        "created": "2015-05-12T15:50:38Z"
    }
}
```

DeleteResult

metadata *Metadata* Metadata of the deleted object.

ERRORS Example: too_many_write_operations

```
{
    "error_summary": "too_many_write_operations/...",
    "error": {
        ".tag": "too_many_write_operations"
    }
}
```

Example: too_many_files

```
{
    "error_summary": "too_many_files/...",
    "error": {
        ".tag": "too_many_files"
    }
}
```

Example: other

```
{
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

DeleteError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path_lookup *LookupError*

path_write *WriteError*

too_many_write_operations *Void* There are too many write operations in user's Dropbox. Please retry this request.

too_many_files *Void* There are too many files in one request. Please retry with fewer files.

/delete_batch

VERSION

1 ▾

DESCRIPTION

Delete multiple files/folders at once.

This route is asynchronous, which returns a job ID immediately and runs the delete batch asynchronously. Use [delete_batch/check](#) to check the job status.

URL STRUCTURE

https://api.dropboxapi.com/2/files/delete_batch

AUTHENTICATION

[User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/delete_batch \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"entries\": [{\"path\": \"/Homework/math/Prime_Numbers.txt\"}]}"
```



≡

```
        "path": "/Homework/math/Prime_Numbers.txt"
    }
}
```

DeleteBatchArg

entries List of (DeleteArg)

RETURNS Example: complete

```
{
    ".tag": "complete",
    "entries": [
        {
            ".tag": "success",
            "metadata": {
                ".tag": "file",
                "name": "Prime_Numbers.txt",
                "id": "id:a4ayc_80_0EAAAAAAAAXw",
                "client_modified": "2015-05-12T15:50:38Z",
                "server_modified": "2015-05-12T15:50:38Z",
                "rev": "a1c10ce0dd78",
                "size": 7212,
                "path_lower": "/homework/math/prime_numbers.txt",
                "path_display": "/Homework/math/Prime_Numbers.txt",
                "sharing_info": {
                    "read_only": true,
                    "parent_shared_folder_id": "84528192421",
                    "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"
                },
                "is_downloadable": true,
                "property_groups": [
                    {
                        "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",
                        "fields": [
                            {
                                "name": "Security Policy",
                                "value": "Confidential"
                            }
                        ]
                    }
                ],
                "has_explicit_shared_members": false,
                "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",
                "file_lock_info": {
                    "is_lockholder": true,
                    "lockholder_name": "Imaginary User",
                    "created": "2015-05-12T15:50:38Z"
                }
            }
        }
    ]
}
```

Example: async_job_id

```
{
    ".tag": "async_job_id",
    "async_job_id": "34g93hh34h04y384084"
}
```

Example: other

```
{
    ".tag": "other"
}
```

DeleteBatchLaunch (open union)

Result returned by [delete_batch](#) that may either launch an asynchronous job or complete synchronously. The value will be one of the following datatypes. New values may be introduced as our API evolves.

async_job_id String(*min_length=1*) This response indicates that the processing is asynchronous. The string is an id that can be used to obtain the status of the asynchronous job.

complete DeleteBatchResult

ERRORS No errors.

/delete_batch/check

VERSION

1 ▾

DESCRIPTION Returns the status of an asynchronous job for [delete_batch](#). If success, it returns list of result for each entry.

URL STRUCTURE

https://api.dropboxapi.com/2/files/delete_batch/check



FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/delete_batch/check \
  -header "Authorization: Bearer <get access token>" \
  -header "Content-Type: application/json" \
  -data "{\"async_job_id\": \"34g93hh34h04y384084\"}"
```

PARAMETERS

```
{  
  "async_job_id": "34g93hh34h04y384084"  
}
```

PollArg

Arguments for methods that poll the status of an asynchronous job. This datatype comes from an imported namespace originally defined in the `async` namespace.

async_job_id *String(min_length=1)* Id of the asynchronous job. This is the value of a response returned from the method that launched the job.

RETURNS

```
{  
  ".tag": "complete",  
  "entries": [  
    {  
      ".tag": "success",  
      "metadata": {  
        ".tag": "file",  
        "name": "Prime_Numbers.txt",  
        "id": "id:a4ayc_80_0EAAAAAAAAXw",  
        "client_modified": "2015-05-12T15:50:38Z",  
        "server_modified": "2015-05-12T15:50:38Z",  
        "rev": "a1c10ce0dd78",  
        "size": 7212,  
        "path_lower": "/homework/math/prime_numbers.txt",  
        "path_display": "/Homework/math/Prime_Numbers.txt",  
        "sharing_info": {  
          "read_only": true,  
          "parent_shared_folder_id": "84528192421",  
          "modified_by": "dbid:AAH4f99T0taONib-OurWxbNQ6ywGRopQngc"  
        },  
        "is_downloadable": true,  
        "property_groups": [  
          {  
            "template_id": "ptid:1a5n2i6d30YEAFFFFFFAYa",  
            "fields": [  
              {  
                "name": "Security Policy",  
                "value": "Confidential"  
              }  
            ]  
          }  
        ],  
        "has_explicit_shared_members": false,  
        "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",  
        "file_lock_info": {  
          "is_lockholder": true,  
          "lockholder_name": "Imaginary User",  
          "created": "2015-05-12T15:50:38Z"  
        }  
      }  
    }  
  ]  
}
```

Example: in_progress

```
{  
  ".tag": "in_progress"  
}
```

Example: other

```
{  
  ".tag": "other"  
}
```

DeleteBatchJobStatus (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

in_progress *Void* The asynchronous job is still in progress.

complete *DeleteBatchResult* The batch delete has finished.

failed *DeleteBatchError* The batch delete has failed.

ERRORS

Example: invalid_async_job_id

```
{  
  "error_summary": "invalid_async_job_id/...",  
  "error": {  
    ".tag": "invalid_async_job_id"  
  }  
}
```



≡



Example: internal_error

```
{  
    "error_summary": "internal_error/...",  
    "error": {  
        ".tag": "internal_error"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

PollError (*open union*)

Error returned by methods for polling the status of asynchronous job. This datatype comes from an imported namespace originally defined in the `async` namespace. The value will be one of the following datatypes. New values may be introduced as our API evolves.

invalid_async_job_id *Void* The job ID is invalid.

internal_error *Void* Something went wrong with the job on Dropbox's end. You'll need to verify that the action you were taking succeeded, and if not, try again. This should happen very rarely.

/download

VERSION

1

DESCRIPTION Download a file from a user's Dropbox.

URL STRUCTURE

<https://content.dropboxapi.com/2/files/download>

AUTHENTICATION [User Authentication, Dropbox-API-Select-Admin \(Whole Team\)](#)

ENDPOINT FORMAT

[Content-download](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://content.dropboxapi.com/2/files/download \  
    --header "Authorization: Bearer <get access token>" \  
    --header "Dropbox-API-Arg: {\"path\": \"/Homework/math/Prime_Numbers.txt\"}"
```

PARAMETERS

```
{  
    "path": "/Homework/math/Prime_Numbers.txt"  
}
```

Example: id

```
{  
    "path": "id:a4ayc_80_0EAAAAAAAAYa"
```

Example: rev

```
{  
    "path": "rev:a1c10ce0dd78"
```

DownloadArg

path *String(pattern="(.|[\\r\\n])*|id:*)|(rev:[0-9a-f]{9,})|(ns:[0-9]+(.*)?")* The path of the file to download.

rev *[deprecated] String(min_length=9, pattern="[0-9a-f]+")?* Field is deprecated. Please specify revision in path instead. This field is optional.

RETURNS

```
{  
    "name": "Prime_Numbers.txt",  
    "id": "id:a4ayc_80_0EAAAAAAAAXw",  
    "client_modified": "2015-05-12T15:50:38Z",  
    "server_modified": "2015-05-12T15:50:38Z",  
    "rev": "a1c10ce0dd78",  
    "size": 7212,  
    "path_lower": "/homework/math/prime_numbers.txt",  
    "path_display": "/Homework/math/Prime_Numbers.txt",  
    "sharing_info": {  
        "read_only": true,  
    }  
}
```



```
        "property_groups": [
            {
                "template_id": "ptid:1a5n2i6d30YEAAAAAAAAYa",
                "fields": [
                    {
                        "name": "Security Policy",
                        "value": "Confidential"
                    }
                ]
            }
        ],
        "has_explicit_shared_members": false,
        "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",
        "file_lock_info": {
            "is_lockholder": true,
            "lockholder_name": "Imaginary User",
            "created": "2015-05-12T15:50:38Z"
        }
    }
}
```

FileMetadata

name *String* The last component of the path (including extension). This never contains a slash.

id *String(min_length=1)* A unique identifier for the file.

client_modified *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")*

For files, this is the modification time set by the desktop client when the file was added to Dropbox. Since this time is not verified (the Dropbox server stores whatever the desktop client sends up), this should only be used for display purposes (such as sorting) and not, for example, to determine if a file has changed or not.

server_modified *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")*
The last time the file was modified on Dropbox.

rev *String(min_length=9, pattern="[0-9a-f]+")* A unique identifier for the current revision of a file. This field is the same rev as elsewhere in the API and can be used to detect changes and avoid conflicts.

size *UInt64* The file size in bytes.

path_lower *String?* The lowercased full path in the user's Dropbox. This always starts with a slash. This field will be null if the file or folder is not mounted. This field is optional.

path_display *String?* The cased path to be used for display purposes only. In rare instances the casing will not correctly match the user's filesystem, but this behavior will match the path provided in the Core API v1, and at least the last path component will have the correct casing. Changes to only the casing of paths won't be returned by [list_folder/continue](#). This field will be null if the file or folder is not mounted. This field is optional.

parent_shared_folder_id [deprecated] *String(pattern="-_0-9a-zA-Z:j+")?* Field is deprecated. Please use FileSharingInfo.parent_shared_folder_id or FolderSharingInfo.parent_shared_folder_id instead. This field is optional.

media_info *MediaInfo?* Additional information if the file is a photo or video. This field will not be set on entries returned by [list_folder](#), [list_folder/continue](#), or [get_thumbnail_batch](#), starting December 2, 2019. This field is optional.

symlink_info *SymlinkInfo?* Set if this file is a symlink. This field is optional.

sharing_info *FileSharingInfo?* Set if this file is contained in a shared folder. This field is optional.

is_downloadable *Boolean* If true, file can be downloaded directly; else the file must be exported. The default for this field is True.

export_info *ExportInfo?* Information about format this file can be exported to. This field must be set if is_downloadable is set to false. This field is optional.

property_groups *List of (PropertyGroup)?* Additional information if the file has custom properties with the property template specified. This field is optional.

has_explicit_shared_members *Boolean?* This flag will only be present if include_has_explicit_shared_members is true in [list_folder](#) or [get_metadata](#). If this flag is present, it will be true if this file has any explicit shared members. This is different from sharing_info in that this could be true in the case where a file has explicit members but is not contained within a shared folder. This field is optional.

content_hash *String(min_length=64, max_length=64)?* A hash of the file content. This field can be used to verify data integrity. For more information see our [Content hash](#) page. This field is optional.

file_lock_info *FileLockMetadata?* If present, the metadata associated with the file's current lock. This field is optional.



```
        ".tag": "unsupported_file"
    }
```

Example: other

```
{
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

DownloadError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path *LookupError*

unsupported_file *Void* This file type cannot be downloaded directly; use [export](#) instead.

/download_zip

VERSION

1

DESCRIPTION Download a folder from the user's Dropbox, as a zip file. The folder must be less than 20 GB in size and have fewer than 10,000 total files. The input cannot be a single file. Any single file must be less than 4GB in size.

URL STRUCTURE

```
https://content.dropboxapi.com/2/files/download_zip
```

AUTHENTICATION

User Authentication

ENDPOINT FORMAT

Content-download

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://content.dropboxapi.com/2/files/download_zip \
--header "Authorization: Bearer <get access token>" \
--header "Dropbox-API-Arg: {\"path\": \"/Homework/math\"}"
```

PARAMETERS

```
{
    "path": "/Homework/math"
}
```

Example: id

```
{
    "path": "id:a4ayc_80_0EAAAAAAAAYa"
}
```

Example: rev

```
{
    "path": "rev:a1c10ce0dd78"
}
```

DownloadZipArg

path *String(pattern="/(.|[r\n])*|id:*)|(rev:[0-9a-f]{9,})|(ns:[0-9]+|(.*)?")* The path of the folder to download.

RETURNS

```
{
    "metadata": {
        "name": "math",
        "id": "id:a4ayc_80_0EAAAAAAAAXz",
        "path_lower": "/homework/math",
        "path_display": "/Homework/math",
        "sharing_info": {
            "read_only": false,
            "parent_shared_folder_id": "84528192421",
            "traverse_only": false,
            "no_access": false
        },
        "property_groups": [
            {
                "template_id": "ptid:la5n2i6d30YEAFFFFFFAYa",
                "fields": [
                    {
                        "name": "Security Policy",
                        "value": "Confidential"
                    }
                ]
            }
        ]
    }
}
```



≡



DownloadZipResult metadata FolderMetadata

ERRORS Example: too_large

```
{  
    "error_summary": "too_large/...",  
    "error": {  
        ".tag": "too_large"  
    }  
}
```

Example: too_many_files

```
{  
    "error_summary": "too_many_files/...",  
    "error": {  
        ".tag": "too_many_files"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

DownloadZipError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path LookupError

too_large Void The folder or a file is too large to download.

too_many_files Void The folder has too many files to download.

/export

VERSION

1 ▾

PREVIEW - may change or disappear without notice

DESCRIPTION Export a file from a user's Dropbox. This route only supports exporting files that cannot be downloaded directly and whose ExportResult.file_metadata has ExportInfo.export_as populated.

URL STRUCTURE

<https://content.dropboxapi.com/2/files/export>

AUTHENTICATION [User Authentication, Dropbox-API-Select-Admin \(Whole Team\)](#)

ENDPOINT FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://content.dropboxapi.com/2/files/export \  
    --header "Authorization: Bearer <get access token>" \  
    --header "Dropbox-API-Arg: {\"path\": \"/Homework/math/Prime_Numbers.gsheet\"}"
```

PARAMETERS

```
{  
    "path": "/Homework/math/Prime_Numbers.gsheet"  
}
```

Example: id

```
{  
    "path": "id:a4ayc_80_0EAAAAAAAAYa"  
}
```

ExportArg

path String(pattern="/(.|[r|n])*|id:.*|(rev:[0-9a-f]{9,})|(ns:[0-9]+(.*)?)"") The path of the file to be exported.

RETURNS

```
{  
    "export_metadata": {  
        "name": "Prime_Numbers.xlsx",  
        "size": 7189,  
        "export_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855"  
    },  
    "file_metadata": {  
        "name": "Prime_Numbers.txt",  
    }  
}
```



```
        "size": 7212,
        "path_lower": "/homework/math/prime_numbers.txt",
        "path_display": "/Homework/math/Prime_Numbers.txt",
        "sharing_info": {
            "read_only": true,
            "parent_shared_folder_id": "84528192421",
            "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"
        },
        "is_downloadable": true,
        "property_groups": [
            {
                "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",
                "fields": [
                    {
                        "name": "Security Policy",
                        "value": "Confidential"
                    }
                ]
            }
        ],
        "has_explicit_shared_members": false,
        "content_hash": "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",
        "file_lock_info": {
            "is_lockholder": true,
            "lockholder_name": "Imaginary User",
            "created": "2015-05-12T15:50:38Z"
        }
    }
}
```

ExportResult

export_metadata *ExportMetadata* Metadata for the exported version of the file.

file_metadata *FileMetadata* Metadata for the original file.

ERRORS Example: non_exportable

```
{
    "error_summary": "non_exportable/",
    "error": {
        ".tag": "non_exportable"
    }
}
```

Example: retry_error

```
{
    "error_summary": "retry_error/",
    "error": {
        ".tag": "retry_error"
    }
}
```

Example: other

```
{
    "error_summary": "other/",
    "error": {
        ".tag": "other"
    }
}
```

ExportError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path *LookupError*

non_exportable *Void* This file type cannot be exported. Use [download](#) instead.

retry_error *Void* The exportable content is not yet available. Please retry later.

/get_file_lock_batch

VERSION

1

PREVIEW - may change or disappear without notice

DESCRIPTION Return the lock metadata for the given list of paths.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/files/get_file_lock_batch

AUTHENTICATION [User Authentication](#)

RPC



≡

ch \

```
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"entries\": [{\"path\": \"/John Doe/sample/test.pdf\"}]}"
```

PARAMETERS

```
{  
    "entries": [  
        {  
            "path": "/John Doe/sample/test.pdf"  
        }  
    ]  
}
```

LockFileBatchArg

entries *List of (LockFileArg)* List of 'entries'. Each 'entry' contains a path of the file which will be locked or queried. Duplicate path arguments in the batch are considered only once.

RETURNS

```
{  
    "entries": [  
        {  
            ".tag": "success",  
            "metadata": {  
                ".tag": "file",  
                "name": "Prime_Numbers.txt",  
                "id": "id:a4ayc_80_0EAAAAAAAAXw",  
                "client_modified": "2015-05-12T15:50:38Z",  
                "server_modified": "2015-05-12T15:50:38Z",  
                "rev": "alc10ce0dd78",  
                "size": 7212,  
                "path_lower": "/homework/math/prime_numbers.txt",  
                "path_display": "/Homework/math/Prime_Numbers.txt",  
                "sharing_info": {  
                    "read_only": true,  
                    "parent_shared_folder_id": "84528192421",  
                    "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"  
                },  
                "is_downloadable": true,  
                "property_groups": [  
                    {  
                        "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",  
                        "fields": [  
                            {  
                                "name": "Security Policy",  
                                "value": "Confidential"  
                            }  
                        ]  
                    }  
                ],  
                "has_explicit_shared_members": false,  
                "content_hash": "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",  
                "file_lock_info": {  
                    "is_lockholder": true,  
                    "lockholder_name": "Imaginary User",  
                    "created": "2015-05-12T15:50:38Z"  
                }  
            },  
            "lock": {  
                "content": {  
                    ".tag": "single_user",  
                    "created": "2015-05-12T15:50:38Z",  
                    "lock_holder_account_id": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc",  
                    "lock_holder_team_id": "dbtid:1234abcd"  
                }  
            }  
        }  
    ]  
}
```

LockFileBatchResult

entries *List of (LockFileResultEntry)* Each Entry in the 'entries' will have '.tag' with the operation status (e.g. success), the metadata for the file and the lock state after the operation.

ERRORS Example: too_many_write_operations

```
{  
    "error_summary": "too_many_write_operations/...",  
    "error": {  
        ".tag": "too_many_write_operations"  
    }  
}
```

Example: too_many_files

```
{  
    "error_summary": "too_many_files/...",  
    "error": {  
        ".tag": "too_many_files"  
    }  
}
```

Example: no_write_permission



≡

```
        ".tag": "no_write_permission"
    }
```

Example: cannot_be_locked

```
{
    "error_summary": "cannot_be_locked/...",
    "error": {
        ".tag": "cannot_be_locked"
    }
}
```

Example: file_not_shared

```
{
    "error_summary": "file_not_shared/...",
    "error": {
        ".tag": "file_not_shared"
    }
}
```

Example: internal_error

```
{
    "error_summary": "internal_error/...",
    "error": {
        ".tag": "internal_error"
    }
}
```

Example: other

```
{
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

LockFileError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path_lookup *LookupError* Could not find the specified resource.

too_many_write_operations *Void* There are too many write operations in user's Dropbox. Please retry this request.

too_many_files *Void* There are too many files in one request. Please retry with fewer files.

no_write_permission *Void* The user does not have permissions to change the lock state or access the file.

cannot_be_locked *Void* Item is a type that cannot be locked.

file_not_shared *Void* Requested file is not currently shared.

lock_conflict *LockConflictError* The user action conflicts with an existing lock on the file.

internal_error *Void* Something went wrong with the job on Dropbox's end. You'll need to verify that the action you were taking succeeded, and if not, try again. This should happen very rarely.

/get_metadata

VERSION

1

DESCRIPTION

Returns the metadata for a file or folder.

Note: Metadata for the root folder is unsupported.

URL STRUCTURE

https://api.dropboxapi.com/2/files/get_metadata

AUTHENTICATION

[User Authentication, Dropbox-API-Select-Admin \(Whole Team\)](#)

ENDPOINT
FORMAT

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/files/get_metadata \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"path\": \"/Homework/math\", \"include_media_info\": false, \"include_deleted\": false, \"include_has_ex\"}
```

PARAMETERS

```
{
    "path": "/Homework/math",
```



Example: id

```
{  
    "path": "id:a4ayc_80_0EAAAAAAAAYa",  
    "include_media_info": false,  
    "include_deleted": false,  
    "include_has_explicit_shared_members": false  
}
```

Example: rev

```
{  
    "path": "rev:alc10ce0dd78",  
    "include_media_info": false,  
    "include_deleted": false,  
    "include_has_explicit_shared_members": false  
}
```

GetMetadataArg

path *String(pattern="/(.|[r\nJ]*|id:*)|(rev:[0-9a-f]{9,})|(ns:[0-9]+(.*)?)")* The path of a file or folder on Dropbox.

include_media_info *Boolean* If true, FileMetadata.media_info is set for photo and video. The default for this field is False.

include_deleted *Boolean* If true, DeletedMetadata will be returned for deleted file or folder, otherwise LookupError.not_found will be returned. The default for this field is False.

include_has_explicit_shared_members *Boolean* If true, the results will include a flag for each file indicating whether or not that file has any explicit members. The default for this field is False.

include_property_groups *TemplateFilterBase?* If set to a valid list of template IDs, FileMetadata.property_groups is set if there exists property data associated with the file and each of the listed templates. This field is optional.

RETURNS

```
{  
    ".tag": "file",  
    "name": "Prime_Numbers.txt",  
    "id": "id:a4ayc_80_0EAAAAAAAAXw",  
    "client_modified": "2015-05-12T15:50:38Z",  
    "server_modified": "2015-05-12T15:50:38Z",  
    "rev": "alc10ce0dd78",  
    "size": 7212,  
    "path_lower": "/homework/math/prime_numbers.txt",  
    "path_display": "/Homework/math/Prime_Numbers.txt",  
    "sharing_info": {  
        "read_only": true,  
        "parent_shared_folder_id": "84528192421",  
        "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"  
    },  
    "is_downloadable": true,  
    "property_groups": [  
        {  
            "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",  
            "fields": [  
                {  
                    "name": "Security Policy",  
                    "value": "Confidential"  
                }  
            ]  
        }  
    ],  
    "has_explicit_shared_members": false,  
    "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",  
    "file_lock_info": {  
        "is_lockholder": true,  
        "lockholder_name": "Imaginary User",  
        "created": "2015-05-12T15:50:38Z"  
    }  
}
```

```
{  
    ".tag": "folder",  
    "name": "math",  
    "id": "id:a4ayc_80_0EAAAAAAAAXz",  
    "path_lower": "/homework/math",  
    "path_display": "/Homework/math",  
    "sharing_info": {  
        "read_only": false,  
        "parent_shared_folder_id": "84528192421",  
        "traverse_only": false,  
        "no_access": false  
    },  
    "property_groups": [  
        {  
            "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",  
            "fields": [  
                {  
                    "name": "Security Policy",  
                    "value": "Confidential"  
                }  
            ]  
        }  
    ]  
}
```



≡



Metadata (*datatype with subtypes*)

Metadata for a file or folder. This datatype will be one of the following subtypes:

file *FileMetadata*
folder *FolderMetadata*
deleted *DeletedMetadata*

ERRORS

GetMetadataError (*union*)

The value will be one of the following datatypes:

path *LookupError*

/get_preview

VERSION

1

DESCRIPTION

Get a preview for a file.

Currently, PDF previews are generated for files with the following extensions: .ai, .doc, .docm, .docx, .eps, .gdoc, .gslides, .odp, .odt, .pps, .ppsm, .ppsx, .ppt, .pptm, .pptx, .rtf.

HTML previews are generated for files with the following extensions: .csv, .ods, .xls, .xlsm, .gsheet, .xlsx.

Other formats will return an unsupported extension error.

URL STRUCTURE

https://content.dropboxapi.com/2/files/get_preview

AUTHENTICATION

[User Authentication, Dropbox-API-Select-Admin \(Whole Team\)](#)

ENDPOINT FORMAT

[Content-download](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://content.dropboxapi.com/2/files/get_preview \
    --header "Authorization: Bearer <get access token>" \
    --header "Dropbox-API-Arg: {\"path\": \"/word.docx\"}"
```

PARAMETERS

```
{
  "path": "/word.docx"
}
```

Example: id

```
{
  "path": "id:a4ayc_80_0EAAAAAAAAYa"
```

Example: rev

```
{
  "path": "rev:alc10ce0dd78"
```

PreviewArg

path *String(pattern="/(.|[\r\n])*|id:*)|(rev:[0-9a-f]{9,})|(ns:[0-9]+(.*)?)"* The path of the file to preview.

rev *[deprecated] String(min_length=9, pattern="[0-9a-f]+")?* Field is deprecated. Please specify revision in path instead. This field is optional.

RETURNS

```
{
  "name": "Prime_Numbers.txt",
  "id": "id:a4ayc_80_0EAAAAAAAAXw",
  "client_modified": "2015-05-12T15:50:38Z",
  "server_modified": "2015-05-12T15:50:38Z",
  "rev": "alc10ce0dd78",
  "size": 7212,
  "path_lower": "/homework/math/prime_numbers.txt",
  "path_display": "/Homework/math/Prime_Numbers.txt",
  "sharing_info": {
    "read_only": true,
    "parent_shared_folder_id": "84528192421",
    "modified_by": "dbid:AAH4f99T0ta0N1b-OurWxbNQ6ywGRopQngc"
  },
  "is_downloadable": true,
  "property_groups": [
    {
      "template_id": "ptid:1a5n2i6d30YEAFFFFFFA",
      "fields": [
        {
          "name": "Security Policy",
          "value": "Confidential"
        }
      ]
    }
  ]
}
```



134ca495991b7852b855",

```
        "file_lock_info": {  
            "is_lockholder": true,  
            "lockholder_name": "Imaginary User",  
            "created": "2015-05-12T15:50:38Z"  
        }  
    }
```

FileMetadata

name *String* The last component of the path (including extension). This never contains a slash.

id *String(min_length=1)* A unique identifier for the file.

client_modified *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")* For files, this is the modification time set by the desktop client when the file was added to Dropbox. Since this time is not verified (the Dropbox server stores whatever the desktop client sends up), this should only be used for display purposes (such as sorting) and not, for example, to determine if a file has changed or not.

server_modified *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")* The last time the file was modified on Dropbox.

rev *String(min_length=9, pattern="[0-9a-f]+")* A unique identifier for the current revision of a file. This field is the same rev as elsewhere in the API and can be used to detect changes and avoid conflicts.

size *UInt64* The file size in bytes.

path_lower *String?* The lowercased full path in the user's Dropbox. This always starts with a slash. This field will be null if the file or folder is not mounted. This field is optional.

path_display *String?* The cased path to be used for display purposes only. In rare instances the casing will not correctly match the user's filesystem, but this behavior will match the path provided in the Core API v1, and at least the last path component will have the correct casing. Changes to only the casing of paths won't be returned by [list_folder/continue](#). This field will be null if the file or folder is not mounted. This field is optional.

parent_shared_folder_id (deprecated) *String(pattern="[-_0-9a-zA-Z:]")?* Field is deprecated. Please use FileSharingInfo.parent_shared_folder_id or FolderSharingInfo.parent_shared_folder_id instead. This field is optional.

media_info *MediaInfo?* Additional information if the file is a photo or video. This field will not be set on entries returned by [list_folder](#), [list_folder/continue](#), or [get_thumbnail_batch](#), starting December 2, 2019. This field is optional.

symlink_info *SymlinkInfo?* Set if this file is a symlink. This field is optional.

sharing_info *FileSharingInfo?* Set if this file is contained in a shared folder. This field is optional.

is_downloadable *Boolean* If true, file can be downloaded directly; else the file must be exported. The default for this field is True.

export_info *ExportInfo?* Information about format this file can be exported to. This field must be set if is_downloadable is set to false. This field is optional.

property_groups *List of (PropertyGroup)?* Additional information if the file has custom properties with the property template specified. This field is optional.

has_explicit_shared_members *Boolean?* This flag will only be present if include_has_explicit_shared_members is true in [list_folder](#) or [get_metadata](#). If this flag is present, it will be true if this file has any explicit shared members. This is different from sharing_info in that this could be true in the case where a file has explicit members but is not contained within a shared folder. This field is optional.

content_hash *String(min_length=64, max_length=64)?* A hash of the file content. This field can be used to verify data integrity. For more information see our [Content hash](#) page. This field is optional.

file_lock_info *FileLockMetadata?* If present, the metadata associated with the file's current lock. This field is optional.

ERRORS

Example: in_progress

```
{  
    "error_summary": "in_progress/...",  
    "error": {  
        ".tag": "in_progress"  
    }  
}
```

Example: unsupported_extension



≡

```
        ".tag": "unsupported_extension"
    }
```

Example: unsupported_content

```
{
    "error_summary": "unsupported_content/...",
    "error": {
        ".tag": "unsupported_content"
    }
}
```

PreviewError (union)

The value will be one of the following datatypes:

path *LookupError* An error occurs when downloading metadata for the file.

in_progress *Void* This preview generation is still in progress and the file is not ready for preview yet.

unsupported_extension *Void* The file extension is not supported preview generation.

unsupported_content *Void* The file content is not supported for preview generation.

/get_temporary_link

VERSION

1

DESCRIPTION Get a temporary link to stream content of a file. This link will expire in four hours and afterwards you will get 410 Gone. This URL should not be used to display content directly in the browser. The Content-Type of the link is determined automatically by the file's mime type.

URL STRUCTURE

https://api.dropboxapi.com/2/files/get_temporary_link

AUTHENTICATION

[User Authentication](#)

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/files/get_temporary_link \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"path\": \"/video.mp4\"}"
```

PARAMETERS

```
{
    "path": "/video.mp4"
}
```

GetTemporaryLinkArg

path *String(pattern="/(.|[r\nJ])*|id:*)|(rev:[0-9a-f]{9,})|(ns:[0-9]+(.*)?")* The path to the file you want a temporary link to.

RETURNS

```
{
    "metadata": {
        "name": "Prime_Numbers.txt",
        "id": "id:a4ayc_80_0EAAAAAAAAXw",
        "client_modified": "2015-05-12T15:50:38Z",
        "server_modified": "2015-05-12T15:50:38Z",
        "rev": "alc10ce0dd78",
        "size": 7212,
        "path_lower": "/homework/math/prime_numbers.txt",
        "path_display": "/Homework/math/Prime_Numbers.txt",
        "sharing_info": {
            "read_only": true,
            "parent_shared_folder_id": "84528192421",
            "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"
        },
        "is_downloadable": true,
        "property_groups": [
            {
                "template_id": "ptid:la5n2i6d30YEAFFFFFFAYa",
                "fields": [
                    {
                        "name": "Security Policy",
                        "value": "Confidential"
                    }
                ]
            }
        ],
        "has_explicit_shared_members": false,
        "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",
        "file_lock_info": {}
    }
}
```



≡

```
    },
    "link": "https://dl.dropboxusercontent.com/apitl/1/YXNkZmFzZGcyMzQyMzI0NjU2NDU2NDU2"
```

GetTemporaryLinkResult

metadata *FileMetadata* Metadata of the file.

link *String* The temporary link which can be used to stream content of the file.

ERRORS Example: email_not_verified

```
{
  "error_summary": "email_not_verified/...",
  "error": {
    ".tag": "email_not_verified"
  }
}
```

Example: unsupported_file

```
{
  "error_summary": "unsupported_file/...",
  "error": {
    ".tag": "unsupported_file"
  }
}
```

Example: other

```
{
  "error_summary": "other/...",
  "error": {
    ".tag": "other"
  }
}
```

GetTemporaryLinkError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path *LookupError*

email_not_verified *Void* The user's email address needs to be verified to use this functionality.

unsupported_file *Void* Cannot get temporary link to this file type; use [export](#) instead.

/get_temporary_upload_link

VERSION

1 ▾

DESCRIPTION Get a one-time use temporary upload link to upload a file to a Dropbox location.

This endpoint acts as a delayed [upload](#). The returned temporary upload link may be used to make a POST request with the data to be uploaded. The upload will then be performed with the CommitInfo previously provided to [get_temporary_upload_link](#) but evaluated only upon consumption. Hence, errors stemming from invalid CommitInfo with respect to the state of the user's Dropbox will only be communicated at consumption time. Additionally, these errors are surfaced as generic HTTP 409 Conflict responses, potentially hiding issue details. The maximum temporary upload link duration is 4 hours. Upon consumption or expiration, a new link will have to be generated. Multiple links may exist for a specific upload path at any given time.

The POST request on the temporary upload link must have its Content-Type set to "application/octet-stream".

Example temporary upload link consumption request:

```
curl -X POST https://dl.dropboxusercontent.com/apitl/1/bNi2uLYF51cVBND
--header "Content-Type: application/octet-stream"
--data-binary @local_file.txt
```

A successful temporary upload link consumption request returns the content hash of the uploaded data in JSON format.

Example successful temporary upload link consumption response:
{"content-hash": "599d71033d700ac892a0e48fa61b125d2f5994"}

An unsuccessful temporary upload link consumption request returns any of the following status codes:



ain or

HTTP 409 Conflict: The temporary upload link does not exist or is currently unavailable, the upload failed, or another error happened.

HTTP 410 Gone: The temporary upload link is expired or consumed.

Example unsuccessful temporary upload link consumption response:
Temporary upload link has been recently consumed.

URL STRUCTURE

```
https://api.dropboxapi.com/2/files/get_temporary_upload_link
```

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/get_temporary_upload_link \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"commit_info\": {\"path\": \"/Homework/math/Matrices.txt\"}, \"mode\": \"add\", \"autorename\": true, \"strict_conflict\": false}
```

PARAMETERS

```
{ "commit_info": {
    "path": "/Homework/math/Matrices.txt",
    "mode": "add",
    "autorename": true,
    "mute": false,
    "strict_conflict": false
},
"duration": 3600
}
```

[GetTemporaryUploadLinkArg](#)

commit_info *CommitInfo* Contains the path and other optional modifiers for the future upload commit. Equivalent to the parameters provided to [upload](#).

duration *Float64(min=60.0, max=14400.0)* How long before this link expires, in seconds. Attempting to start an upload with this link longer than this period of time after link creation will result in an error. The default for this field is 14400.0.

RETURNS

```
{ "link": "https://dl.dropboxusercontent.com/apitul/1/bNi2uIYF51cVBND"
```

[GetTemporaryUploadLinkResult](#)

link *String* The temporary link which can be used to stream a file to a Dropbox location.

ERRORS

No errors.

[/get_thumbnail](#)

VERSION

2 ▾

DESCRIPTION

Get a thumbnail for a file.

URL STRUCTURE

```
https://content.dropboxapi.com/2/files/get_thumbnail_v2
```

AUTHENTICATION

[User Authentication, Dropbox-API-Select-Admin \(Whole Team\)](#)

ENDPOINT FORMAT

[Content-download](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://content.dropboxapi.com/2/files/get_thumbnail_v2 \
--header "Authorization: Bearer <get access token>" \
--header "Dropbox-API-Arg: {\"resource\": {\".tag\": \"path\", \"path\": \"/a.docx\"}, \"format\": \"jpeg\", \"size\": \"w64h64\"}
```

PARAMETERS

```
{ "resource": {
    ".tag": "path",
    "path": "/a.docx"
},
"format": "jpeg",
"size": "w64h64",
```



≡



1 numonauvzARG

resource *PathOrLink* Information specifying which file to preview.

This could be a path to a file, a shared link pointing to a file, or a shared link pointing to a folder, with a relative path.

format *ThumbnailFormat* The format for the thumbnail image, jpeg (default) or png. For images that are photos, jpeg should be preferred, while png is better for screenshots and digital arts. The default for this union is jpeg.**size** *ThumbnailSize* The size for the thumbnail image. The default for this union is w64h64.**mode** *ThumbnailMode* How to resize and crop the image to achieve the desired size. The default for this union is strict.

RETURNS

```
{  
    "file_metadata": {  
        "name": "Prime_Numbers.txt",  
        "id": "id:a4ayc_80_0EAAAAAAAAXw",  
        "client_modified": "2015-05-12T15:50:38Z",  
        "server_modified": "2015-05-12T15:50:38Z",  
        "rev": "alc10ce0dd78",  
        "size": 7212,  
        "path_lower": "/homework/math/prime_numbers.txt",  
        "path_display": "/Homework/math/Prime_Numbers.txt",  
        "sharing_info": {  
            "read_only": true,  
            "parent_shared_folder_id": "84528192421",  
            "modified_by": "dbid:AAH4f99T0ta0N1b-OurWxbNQ6ywGRopQngc"  
        },  
        "is_downloadable": true,  
        "property_groups": [  
            {  
                "template_id": "ptid:1a5n2i6d30YEAAAAAAAAYa",  
                "fields": [  
                    {  
                        "name": "Security Policy",  
                        "value": "Confidential"  
                    }  
                ]  
            },  
            {  
                "has_explicit_shared_members": false,  
                "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",  
                "file_lock_info": {  
                    "is_lockholder": true,  
                    "lockholder_name": "Imaginary User",  
                    "created": "2015-05-12T15:50:38Z"  
                }  
            }  
        ]  
    }  
}
```

PreviewResult

file_metadata *FileMetadata?* Metadata corresponding to the file received as an argument. Will be populated if the endpoint is called with a path (ReadPath). This field is optional.**link_metadata** *MinimalFileLinkMetadata?* Minimal metadata corresponding to the file received as an argument. Will be populated if the endpoint is called using a shared link (SharedLinkFileInfo). This field is optional.

ERRORS

Example: unsupported_extension

```
{  
    "error_summary": "unsupported_extension/...",  
    "error": {  
        ".tag": "unsupported_extension"  
    }  
}
```

Example: unsupported_image

```
{  
    "error_summary": "unsupported_image/...",  
    "error": {  
        ".tag": "unsupported_image"  
    }  
}
```

Example: conversion_error

```
{  
    "error_summary": "conversion_error/...",  
    "error": {  
        ".tag": "conversion_error"  
    }  
}
```

Example: access_denied

```
{  
    "error_summary": "access_denied/...",  
}
```



Example: not_found

```
{  
    "error_summary": "not_found/...",  
    "error": {  
        ".tag": "not_found"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

ThumbnailV2Error (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path *LookupError* An error occurred when downloading metadata for the image.

unsupported_extension *Void* The file extension doesn't allow conversion to a thumbnail.

unsupported_image *Void* The image cannot be converted to a thumbnail.

conversion_error *Void* An error occurred during thumbnail conversion.

access_denied *Void* Access to this shared link is forbidden.

not_found *Void* The shared link does not exist.

/get_thumbnail_batch

VERSION

1 ▾

DESCRIPTION

Get thumbnails for a list of images. We allow up to 25 thumbnails in a single batch.

This method currently supports files with the following file extensions: jpg, jpeg, png, tiff, tif, gif and bmp. Photos that are larger than 20MB in size won't be converted to a thumbnail.

URL STRUCTURE

https://content.dropboxapi.com/2/files/get_thumbnail_batch

AUTHENTICATION

[User Authentication, Dropbox-API-Select-Admin \(Whole Team\)](#)

ENDPOINT
FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://content.dropboxapi.com/2/files/get_thumbnail_batch \  
--header "Authorization: Bearer <get access token>" \  
--header "Content-Type: application/json" \  
-data "{\"entries\": [{\"path\": \"/image.jpg\", \"format\": \"jpeg\", \"size\": \"w64h64\", \"mode\": \"strict\"}]}
```

PARAMETERS

```
{  
    "entries": [  
        {  
            "path": "/image.jpg",  
            "format": "jpeg",  
            "size": "w64h64",  
            "mode": "strict"  
        }  
    ]  
}
```

[GetThumbnailBatchArg](#)

Arguments for [get_thumbnail_batch](#).

entries *List of (ThumbnailArg)* List of files to get thumbnails.

RETURNS

```
{  
    "entries": [  
        {  
            ".tag": "success",  
            "metadata": {  
                "name": "Prime_Numbers.txt",  
                "id": "id:a4ayc_80_0EAAAAAAAAXw",  
                "client_modified": "2015-05-12T15:50:38Z",  
                "server_modified": "2015-05-12T15:50:38Z",  
                "rev": "alc10ce0dd78",  
                "size": 7212,  
                "path_lower": "/homework/math/prime_numbers.txt",  
                "path_upper": "/homework/math/prime_numbers.txt"  
            }  
        }  
    ]  
}
```



```
        "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"
    },
    "is_downloadable": true,
    "property_groups": [
        {
            "template_id": "ptid:la5n2i6d30YEAFFFFFFAYa",
            "fields": [
                {
                    "name": "Security Policy",
                    "value": "Confidential"
                }
            ]
        }
    ],
    "has_explicit_shared_members": false,
    "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",
    "file_lock_info": {
        "is_lockholder": true,
        "lockholder_name": "Imaginary User",
        "created": "2015-05-12T15:50:38Z"
    },
    "thumbnail": "iVBORw0KGgoAAAANSUhEUgAAAdcAAABrCAMAII="
}
]
```

GetThumbnailBatchResult

entries *List of (GetThumbnailBatchResultEntry)* List of files and their thumbnails.

ERRORS Example: `too_many_files`

```
{
    "error_summary": "too_many_files/...",
    "error": {
        ".tag": "too_many_files"
    }
}
```

Example: `other`

```
{
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

GetThumbnailBatchError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

too_many_files *Void* The operation involves more than 25 files.

/list_folder

VERSION

1

DESCRIPTION

Starts returning the contents of a folder. If the result's `ListFolderResult.has_more` field is true,

call `list_folder/continue` with the returned `ListFolderResult.cursor` to retrieve more entries.

If you're using `ListFolderArg.recursive` set to true to keep a local cache of the contents of a

Dropbox account, iterate through each entry in order and process them as follows to keep your local state in sync:

For each `FileMetadata`, store the new entry at the given path in your local state. If the required parent folders don't exist yet, create them. If there's already something else at the given path, replace it and remove all its children.

For each `FolderMetadata`, store the new entry at the given path in your local state. If the required parent folders don't exist yet, create them. If there's already something else at the given path, replace it but leave the children as they are. Check the new entry's `FolderSharingInfo.read_only` and set all its children's read-only statuses to match.

For each `DeletedMetadata`, if your local state has something at the given path, remove it and all its children. If there's nothing at the given path, ignore this entry.

Note: `auth.RateLimitError` may be returned if multiple `list_folder` or `list_folder/continue` calls with same parameters are made simultaneously by same API app for same user. If your app implements retry logic, please hold off the retry until the previous request finishes.

URL STRUCTURE

https://api.dropboxapi.com/2/files/list_folder

AUTHENTICATION

[User Authentication, Dropbox-API-Select-Admin \(Whole Team\)](#)

ENDPOINT
FORMAT

RPC

EXAMPLE



```
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"path\": \"/Homework/math\", \"recursive\": false, \"include_media_info\": false, \"include_deleted\": 1}
```

PARAMETERS

```
{  
    "path": "/Homework/math",  
    "recursive": false,  
    "include_media_info": false,  
    "include_deleted": false,  
    "include_has_explicit_shared_members": false,  
    "include_mounted_folders": true,  
    "include_non_downloadable_files": true  
}
```

ListFolderArg

path *String(pattern="(.|[\\r\\n])*?|id:.*|(ns:[0-9]+(.*)?)")* A unique identifier for the file.

recursive *Boolean* If true, the list folder operation will be applied recursively to all subfolders and the response will contain contents of all subfolders. The default for this field is False.

include_media_info *deprecated Boolean* Field is deprecated. If true, FileMetadata.media_info is set for photo and video. This parameter will no longer have an effect starting December 2, 2019. The default for this field is False.

include_deleted *Boolean* If true, the results will include entries for files and folders that used to exist but were deleted. The default for this field is False.

include_has_explicit_shared_members *Boolean* If true, the results will include a flag for each file indicating whether or not that file has any explicit members. The default for this field is False.

include_mounted_folders *Boolean* If true, the results will include entries under mounted folders which includes app folder, shared folder and team folder. The default for this field is True.

limit *UInt32(min=1, max=2000)?* The maximum number of results to return per request. Note: This is an approximate number and there can be slightly more entries returned in some cases. This field is optional.

shared_link *SharedLink?* A shared link to list the contents of. If the link is password-protected, the password must be provided. If this field is present, ListFolderArg.path will be relative to root of the shared link. Only non-recursive mode is supported for shared link. This field is optional.

include_property_groups *TemplateFilterBase?* If set to a valid list of template IDs, FileMetadata.property_groups is set if there exists property data associated with the file and each of the listed templates. This field is optional.

include_non_downloadable_files *Boolean* If true, include files that are not downloadable, i.e. Google Docs. The default for this field is True.

RETURNS

```
{  
    "entries": [  
        {  
            ".tag": "file",  
            "name": "Prime_Numbers.txt",  
            "id": "id:a4ayc_80_0EAAAAAAAAXw",  
            "client_modified": "2015-05-12T15:50:38Z",  
            "server_modified": "2015-05-12T15:50:38Z",  
            "rev": "alc10ce0dd78",  
            "size": 7212,  
            "path_lower": "/homework/math/prime_numbers.txt",  
            "path_display": "/Homework/math/Prime_Numbers.txt",  
            "sharing_info": {  
                "read_only": true,  
                "parent_shared_folder_id": "84528192421",  
                "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"  
            },  
            "is_downloadable": true,  
            "property_groups": [  
                {  
                    "template_id": "ptid:1a5n2i6d30YAAAAAAAAYa",  
                    "fields": [  
                        {  
                            "name": "Security Policy",  
                            "value": "Confidential"  
                        }  
                    ]  
                }  
            ],  
            "has_explicit_shared_members": false,  
            "content_hash": "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",  
            "file_lock_info": {  
                "is_lockholder": true,  
                "lockholder_name": "Imaginary User",  
                "created": "2015-05-12T15:50:38Z"  
            }  
        },  
        {  
            ".tag": "folder",  
            "name": "Math",  
            "id": "id:12345678901234567890123456789012",  
            "client_modified": "2015-05-12T15:50:38Z",  
            "server_modified": "2015-05-12T15:50:38Z",  
            "rev": "alc10ce0dd78",  
            "size": 0,  
            "path_lower": "/homework/math/Math",  
            "path_display": "/Homework/math/Math",  
            "sharing_info": {  
                "read_only": true,  
                "parent_shared_folder_id": "84528192421",  
                "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"  
            },  
            "is_downloadable": false,  
            "property_groups": []  
        }  
    ]  
}
```



```
        "path_display": "/Homework/math",
        "sharing_info": {
            "read_only": false,
            "parent_shared_folder_id": "84528192421",
            "traverse_only": false,
            "no_access": false
        },
        "property_groups": [
            {
                "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",
                "fields": [
                    {
                        "name": "Security Policy",
                        "value": "Confidential"
                    }
                ]
            }
        ],
        "cursor": "ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu",
        "has_more": false
    }
}
```

ListFolderResult

entries *List of (Metadata)* The files and (direct) subfolders in the folder.

cursor *String(min_length=1)* Pass the cursor into

[list_folder/continue](#) to see what's changed in the folder since your previous query.

has_more *Boolean* If true, then there are more entries available. Pass the cursor to [list_folder/continue](#) to retrieve the rest.

ERRORS

ListFolderError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path *LookupError*

template_error *TemplateError*

/list_folder/continue

VERSION

1 ▾

DESCRIPTION Once a cursor has been retrieved from [list_folder](#), use this to paginate through all files and retrieve updates to the folder, following the same rules as documented for [list_folder](#).

URL STRUCTURE

https://api.dropboxapi.com/2/files/list_folder/continue

AUTHENTICATION [User Authentication](#), [Dropbox-API-Select-Admin](#) (Whole Team)

ENDPOINT FORMAT

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/list_folder/continue \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"cursor\": \"ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu\"}"
```

PARAMETERS

```
{
    "cursor": "ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu"
}
```

ListFolderContinueArg

cursor *String(min_length=1)* The cursor returned by your last call to [list_folder](#) or [list_folder/continue](#).

RETURNS

```
{
    "entries": [
        {
            ".tag": "file",
            "name": "Prime_Numbers.txt",
            "id": "id:aayc_80_OEAAAAAAAAXw",
            "client_modified": "2015-05-12T15:50:38Z",
            "server_modified": "2015-05-12T15:50:38Z",
            "rev": "a1c10ce0dd78",
            "size": 7212,
            "path_lower": "/homework/math/prime_numbers.txt",
            "path_display": "/Homework/math/Prime_Numbers.txt",
            "sharing_info": {
                "read_only": true,
                "parent_shared_folder_id": "84528192421",
                "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"
            },
            ...
        }
    ]
}
```



```
        "fields": [
            {
                "name": "Security Policy",
                "value": "Confidential"
            }
        ]
    },
    "has_explicit_shared_members": false,
    "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",
    "file_lock_info": {
        "is_lockholder": true,
        "lockholder_name": "Imaginary User",
        "created": "2015-05-12T15:50:38Z"
    }
},
{
    ".tag": "folder",
    "name": "math",
    "id": "id:aayc_80_OEAAAAAAAAXz",
    "path_lower": "/homework/math",
    "path_display": "/Homework/math",
    "sharing_info": {
        "read_only": false,
        "parent_shared_folder_id": "84528192421",
        "traverse_only": false,
        "no_access": false
    },
    "property_groups": [
        {
            "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",
            "fields": [
                {
                    "name": "Security Policy",
                    "value": "Confidential"
                }
            ]
        }
    ],
    "cursor": "ZtkX9_EHj3x7PMkVuFIhwKYXEpwplwyxp9vMKomUhllil9q7eWiAu",
    "has_more": false
}
```

ListFolderResult

entries *List of (Metadata)* The files and (direct) subfolders in the folder.

cursor *String(min_length=1)* Pass the cursor into [list_folder/continue](#) to see what's changed in the folder since your previous query.

has_more *Boolean* If true, then there are more entries available. Pass the cursor to [list_folder/continue](#) to retrieve the rest.

ERRORS Example: reset

```
{
    "error_summary": "reset/...",
    "error": {
        ".tag": "reset"
    }
}
```

Example: other

```
{
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

ListFolderContinueError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path *LookupError*

reset *Void* Indicates that the cursor has been invalidated. Call [list_folder](#) to obtain a new cursor.

/list_folder/get_latest_cursor

VERSION

1

DESCRIPTION

A way to quickly get a cursor for the folder's state. Unlike [list_folder](#), [list_folder/get_latest_cursor](#) doesn't return any entries. This endpoint is for app which only needs to know about new files and modifications and doesn't need to know about files that already exist in Dropbox.



AUTHENTICATION

User Authentication, Dropbox-API-Select-Admin (Whole Team)

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/list_folder/get_latest_cursor \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"path\": \"/Homework/math\", \"recursive\": false, \"include_media_info\": false, \"include_deleted\": true}
```

PARAMETERS

```
{  
  "path": "/Homework/math",  
  "recursive": false,  
  "include_media_info": false,  
  "include_deleted": false,  
  "include_has_explicit_shared_members": false,  
  "include_mounted_folders": true,  
  "include_non_downloadable_files": true  
}
```

ListFolderArg

path *String(pattern="(.|[r|n])*?[id:.*|(ns:[0-9]+(.*)?)"")* A unique identifier for the file.

recursive *Boolean* If true, the list folder operation will be applied recursively to all subfolders and the response will contain contents of all subfolders. The default for this field is False.

include_media_info deprecated *Boolean* Field is deprecated. If true, FileMetadata.media_info is set for photo and video. This parameter will no longer have an effect starting December 2, 2019. The default for this field is False.

include_deleted *Boolean* If true, the results will include entries for files and folders that used to exist but were deleted. The default for this field is False.

include_has_explicit_shared_members *Boolean* If true, the results will include a flag for each file indicating whether or not that file has any explicit members. The default for this field is False.

include_mounted_folders *Boolean* If true, the results will include entries under mounted folders which includes app folder, shared folder and team folder. The default for this field is True.

limit *UInt32(min=1, max=2000)?* The maximum number of results to return per request. Note: This is an approximate number and there can be slightly more entries returned in some cases. This field is optional.

shared_link *SharedLink?* A shared link to list the contents of. If the link is password-protected, the password must be provided. If this field is present, ListFolderArg.path will be relative to root of the shared link. Only non-recursive mode is supported for shared link. This field is optional.

include_property_groups *TemplateFilterBase?* If set to a valid list of template IDs, FileMetadata.property_groups is set if there exists property data associated with the file and each of the listed templates. This field is optional.

include_non_downloadable_files *Boolean* If true, include files that are not downloadable, i.e. Google Docs. The default for this field is True.

RETURNS

```
{  
  "cursor": "ZtkX9_EHj3x7PMkVuFIhwKYXEpwplwyxp9vMKomUhllil9q7eWiAu"  
}
```

ListFolderGetLatestCursorResult

cursor *String(min_length=1)* Pass the cursor into [list_folder/continue](#) to see what's changed in the folder since your previous query.

ERRORS

ListFolderError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path *LookupError*

template_error *TemplateError*

/list_folder/longpoll

VERSION

1 ▾



s. This

endpoint is useful mostly for client-side apps. If you're looking for server-side notifications, check out our [webhooks documentation](#).

URL STRUCTURE

```
https://notify.dropboxapi.com/2/files/list_folder/longpoll
```

AUTHENTICATION

[No Authentication](#), [Dropbox-API-Select-Admin \(Whole Team\)](#)

ENDPOINT FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://notify.dropboxapi.com/2/files/list_folder/longpoll \
--header "Content-Type: application/json" \
--data "{\"cursor\": \"ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu\", \"timeout\": 30}"
```

PARAMETERS

```
{\n    \"cursor\": \"ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu\", \n    \"timeout\": 30\n}
```

ListFolderLongpollArg

cursor *String(min_length=1)* A cursor as returned by [list_folder](#) or [list_folder/continue](#). Cursors retrieved by setting `ListFolderArg.include_media_info` to true are not supported.

timeout *UInt64(min=30, max=480)* A timeout in seconds. The request will block for at most this length of time, plus up to 90 seconds of random jitter added to avoid the thundering herd problem. Care should be taken when using this parameter, as some network infrastructure does not support long timeouts. The default for this field is 30.

RETURNS

```
{\n    \"changes\": true\n}
```

ListFolderLongpollResult

changes *Boolean* Indicates whether new changes are available. If true, call [list_folder/continue](#) to retrieve the changes.

backoff *UInt64?* If present, backoff for at least this many seconds before calling [list_folder/longpoll](#) again. This field is optional.

ERRORS

Example: reset

```
{\n    \"error_summary\": \"reset/...\", \n    \"error\": {\n        \".tag\": \"reset\"\n    }\n}
```

Example: other

```
{\n    \"error_summary\": \"other/...\", \n    \"error\": {\n        \".tag\": \"other\"\n    }\n}
```

ListFolderLongpollError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

reset *Void* Indicates that the cursor has been invalidated. Call [list_folder](#) to obtain a new cursor.

/list_revisions

VERSION

1 ▾

DESCRIPTION

Returns revisions for files based on a file path or a file id. The file path or file id is identified from the latest file entry at the given file path or id. This endpoint allows your app to query either by file path or file id by setting the mode parameter appropriately.

In the `ListRevisionsMode.path` (default) mode, all revisions at the same file path as the latest file entry are returned. If revisions with the same file id are desired, then mode must be set to `ListRevisionsMode.id`. The `ListRevisionsMode.id` mode is useful to retrieve revisions for a given file across moves or renames.



AUTHENTICATION

User Authentication, Dropbox-API-Select-Admin (Whole Team)

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/list_revisions \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"path\": \"/root/word.docx\", \"mode\": \"path\", \"limit\": 10}"
```

PARAMETERS

```
{  
    "path": "/root/word.docx",  
    "mode": "path",  
    "limit": 10  
}
```

ListRevisionsArg

path *String(pattern="/(.|[r\n])*|id:.*|(ns:[0-9]+(.*))")* The path to the file you want to see the revisions of.

mode *ListRevisionsMode* Determines the behavior of the API in listing the revisions for a given file path or id. The default for this union is path.

limit *UInt64(min=1, max=100)* The maximum number of revision entries returned. The default for this field is 10.

RETURNS

```
{  
    "is_deleted": false,  
    "entries": [  
        {  
            "name": "Prime_Numbers.txt",  
            "id": "id:a4ayc_80_OEAAAAAAAAXw",  
            "client_modified": "2015-05-12T15:50:38Z",  
            "server_modified": "2015-05-12T15:50:38Z",  
            "rev": "alc10ce0dd78",  
            "size": 7212,  
            "path_lower": "/homework/math/prime_numbers.txt",  
            "path_display": "/Homework/math/Prime_Numbers.txt",  
            "sharing_info": {  
                "read_only": true,  
                "parent_shared_folder_id": "84528192421",  
                "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"  
            },  
            "is_downloadable": true,  
            "property_groups": [  
                {  
                    "template_id": "ptid:la5n2i6d30YEAFFFFFFAYa",  
                    "fields": [  
                        {  
                            "name": "Security Policy",  
                            "value": "Confidential"  
                        }  
                    ]  
                }  
            ],  
            "has_explicit_shared_members": false,  
            "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",  
            "file_lock_info": {  
                "is_lockholder": true,  
                "lockholder_name": "Imaginary User",  
                "created": "2015-05-12T15:50:38Z"  
            }  
        }  
    ]  
}
```

ListRevisionsResult

is_deleted *Boolean* If the file identified by the latest revision in the response is either deleted or moved.

entries *List of (FileMetadata)* The revisions for the file. Only revisions that are not deleted will show up here.

server_deleted *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")?* The time of deletion if the file was deleted. This field is optional.

ERRORS

ListRevisionsError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path *LookupError*

/lock_file_batch

VERSION

1 ▾

PREVIEW - may change or disappear without notice

DESCRIPTION

Lock the files at the given paths. A locked file will be writable only by the lock holder. A successful response indicates that the file has been locked. Returns a list of the locked file



This endpoint does not support apps with the app folder permission.

URL STRUCTURE

```
https://api.dropboxapi.com/2/files/lock_file_batch
```

AUTHENTICATION

User Authentication

ENDPOINT
FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/lock_file_batch \
  -header "Authorization: Bearer <get access token>" \
  -header "Content-Type: application/json" \
  --data "{\"entries\": [{\"path\": \"/John Doe/sample/test.pdf\"}]}"
```

PARAMETERS

```
{  
    "entries": [  
        {  
            "path": "/John Doe/sample/test.pdf"  
        }  
    ]  
}
```

LockFileBatchArg

entries List of (LockFileArg) List of 'entries'. Each 'entry' contains a path of the file which will be locked or queried. Duplicate path arguments in the batch are considered only once.

RETURNS

```
{  
    "entries": [  
        {  
            ".tag": "success",  
            "metadata": {  
                ".tag": "file",  
                "name": "Prime_Numbers.txt",  
                "id": "id:a4ayc_80_0EAAAAAAAAXw",  
                "client_modified": "2015-05-12T15:50:38Z",  
                "server_modified": "2015-05-12T15:50:38Z",  
                "rev": "alc10ce0dd78",  
                "size": 7212,  
                "path_lower": "/homework/math/prime_numbers.txt",  
                "path_display": "/Homework/math/Prime_Numbers.txt",  
                "sharing_info": {  
                    "read_only": true,  
                    "parent_shared_folder_id": "84528192421",  
                    "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"  
                },  
                "is_downloadable": true,  
                "property_groups": [  
                    {  
                        "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",  
                        "fields": [  
                            {  
                                "name": "Security Policy",  
                                "value": "Confidential"  
                            }  
                        ]  
                    }  
                ],  
                "has_explicit_shared_members": false,  
                "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",  
                "file_lock_info": {  
                    "is_lockholder": true,  
                    "lockholder_name": "Imaginary User",  
                    "created": "2015-05-12T15:50:38Z"  
                }  
            },  
            "lock": {  
                "content": {  
                    ".tag": "single_user",  
                    "created": "2015-05-12T15:50:38Z",  
                    "lock_holder_account_id": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc",  
                    "lock_holder_team_id": "dbtid:1234abcd"  
                }  
            }  
        }  
    ]  
}
```

LockFileBatchResult

entries List of (LockFileResultEntry) Each Entry in the 'entries' will have '.tag' with the operation status (e.g. success), the metadata for the file and the lock state after the operation.

ERRORS

Example: too_many_write_operations

```
{  
    "error_summary": "too_many_write_operations/...",  
    "error": {  
        ".tag": "too_many_write_operations"
```



Example: too_many_files

```
{  
    "error_summary": "too_many_files/...",  
    "error": {  
        ".tag": "too_many_files"  
    }  
}
```

Example: no_write_permission

```
{  
    "error_summary": "no_write_permission/...",  
    "error": {  
        ".tag": "no_write_permission"  
    }  
}
```

Example: cannot_be_locked

```
{  
    "error_summary": "cannot_be_locked/...",  
    "error": {  
        ".tag": "cannot_be_locked"  
    }  
}
```

Example: file_not_shared

```
{  
    "error_summary": "file_not_shared/...",  
    "error": {  
        ".tag": "file_not_shared"  
    }  
}
```

Example: internal_error

```
{  
    "error_summary": "internal_error/...",  
    "error": {  
        ".tag": "internal_error"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

LockFileError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path_lookup *LookupError* Could not find the specified resource.

too_many_write_operations *Void* There are too many write operations in user's Dropbox. Please retry this request.

too_many_files *Void* There are too many files in one request. Please retry with fewer files.

no_write_permission *Void* The user does not have permissions to change the lock state or access the file.

cannot_be_locked *Void* Item is a type that cannot be locked.

file_not_shared *Void* Requested file is not currently shared.

lock_conflict *LockConflictError* The user action conflicts with an existing lock on the file.

internal_error *Void* Something went wrong with the job on Dropbox's end. You'll need to verify that the action you were taking succeeded, and if not, try again. This should happen very rarely.

/move

VERSION

2 ▾

DESCRIPTION Move a file or folder to a different location in the user's Dropbox.
If the source path is a folder all its contents will be moved.

URL STRUCTURE

https://api.dropboxapi.com/2/files/move_v2

AUTHENTICATION



FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/move_v2 \
  -header "Authorization: Bearer <get access token>" \
  -header "Content-Type: application/json" \
  -data "{\"from_path\": \"/Homework/math\", \"to_path\": \"/Homework/algebra\", \"allow_shared_folder\": false, \"\"}
```

PARAMETERS

```
{  
  \"from_path\": \"/Homework/math\",  
  \"to_path\": \"/Homework/algebra\",  
  \"allow_shared_folder\": false,  
  \"autorename\": false,  
  \"allow_ownership_transfer\": false  
}
```

RelocationArg

from_path *String(pattern="(.|[\\r\\n])*|(ns:[0-9]+(\\.?)?|(id:.*))")*
Path in the user's Dropbox to be copied or moved.

to_path *String(pattern="(.|[\\r\\n])*|(ns:[0-9]+(\\.?)?|(id:.*))")* Path in the user's Dropbox that is the destination.

allow_shared_folder *Boolean* If true, [copy](#) will copy contents in shared folder, otherwise RelocationError.cant_copy_shared_folder will be returned if from_path contains shared folder. This field is always true for [move](#). The default for this field is False.

autorename *Boolean* If there's a conflict, have the Dropbox server try to autorename the file to avoid the conflict. The default for this field is False.

allow_ownership_transfer *Boolean* Allow moves by owner even if it would result in an ownership transfer for the content being moved. This does not apply to copies. The default for this field is False.

RETURNS

```
{  
  \"metadata\": {  
    \".tag\": \"file\",  
    \"name\": \"Prime_Numbers.txt\",  
    \"id\": \"id:a4ayc_80_0EAAAAAAAAXw\",  
    \"client_modified\": \"2015-05-12T15:50:38Z\",  
    \"server_modified\": \"2015-05-12T15:50:38Z\",  
    \"rev\": \"alc10ce0dd78\",  
    \"size\": 7212,  
    \"path_lower\": \"/homework/math/prime_numbers.txt\",  
    \"path_display\": \"/Homework/math/Prime_Numbers.txt\",  
    \"sharing_info\": {  
      \"read_only\": true,  
      \"parent_shared_folder_id\": \"84528192421\",  
      \"modified_by\": \"dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc\"  
    },  
    \"is_downloadable\": true,  
    \"property_groups\": [  
      {  
        \"template_id\": \"ptid:la5n2i6d30YEAFFFFFFAYa\",  
        \"fields\": [  
          {  
            \"name\": \"Security Policy\",  
            \"value\": \"Confidential\"  
          }  
        ]  
      }  
    ],  
    \"has_explicit_shared_members\": false,  
    \"content_hash\": \"e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855\",  
    \"file_lock_info\": {  
      \"is_lockholder\": true,  
      \"lockholder_name\": \"Imaginary User\",  
      \"created\": \"2015-05-12T15:50:38Z\"  
    }  
  }  
}
```

RelocationResult

metadata *Metadata* Metadata of the relocated object.

ERRORS

Example: cant_copy_shared_folder

```
{  
  \"error_summary\": \"cant_copy_shared_folder/...\",  
  \"error\": {  
    \".tag\": \"cant_copy_shared_folder\"\br/>  }  
}
```

Example: cant_nest_shared_folder

```
{  
  \"error_summary\": \"cant_nest_shared_folder/...\",  
  \"error\": {  
    \".tag\": \"cant_nest_shared_folder\"\br/>  }  
}
```



```
{  
    "error_summary": "cant_move_folder_into_itself/...",  
    "error": {  
        ".tag": "cant_move_folder_into_itself"  
    }  
}
```

Example: too_many_files

```
{  
    "error_summary": "too_many_files/...",  
    "error": {  
        ".tag": "too_many_files"  
    }  
}
```

Example: duplicated_or_nested_paths

```
{  
    "error_summary": "duplicated_or_nested_paths/...",  
    "error": {  
        ".tag": "duplicated_or_nested_paths"  
    }  
}
```

Example: cant_transfer_ownership

```
{  
    "error_summary": "cant_transfer_ownership/...",  
    "error": {  
        ".tag": "cant_transfer_ownership"  
    }  
}
```

Example: insufficient_quota

```
{  
    "error_summary": "insufficient_quota/...",  
    "error": {  
        ".tag": "insufficient_quota"  
    }  
}
```

Example: internal_error

```
{  
    "error_summary": "internal_error/...",  
    "error": {  
        ".tag": "internal_error"  
    }  
}
```

Example: cant_move_shared_folder

```
{  
    "error_summary": "cant_move_shared_folder/...",  
    "error": {  
        ".tag": "cant_move_shared_folder"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

RelocationError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

from_lookup *LookupError*

from_write *WriteError*

to *WriteError*

cant_copy_shared_folder *Void* Shared folders can't be copied.

cant_nest_shared_folder *Void* Your move operation would result in nested shared folders. This is not allowed.

cant_move_folder_into_itself *Void* You cannot move a folder into itself.

too_many_files *Void* The operation would involve more than 10,000 files and folders.

duplicated_or_nested_paths *Void* There are duplicated/nested paths among RelocationArg.from_path and RelocationArg.to_path.



insufficient_quota *Void* The current user does not have enough space to move or copy the files.

internal_error *Void* Something went wrong with the job on Dropbox's end. You'll need to verify that the action you were taking succeeded, and if not, try again. This should happen very rarely.

can't_move_shared_folder *Void* Can't move the shared folder to the given destination.

/move_batch

VERSION

2

DESCRIPTION Move multiple files or folders to different locations at once in the user's Dropbox.
This route will replace [move_batch:1](#). The main difference is this route will return status for each entry, while [move_batch:1](#) raises failure if any entry fails.
This route will either finish synchronously, or return a job ID and do the async move job in background. Please use [move_batch/check:2](#) to check the job status.

URL STRUCTURE

https://api.dropboxapi.com/2/files/move_batch_v2

AUTHENTICATION [User Authentication, Dropbox-API-Select-Admin \(Team Admin\)](#)

ENDPOINT
FORMAT

EXAMPLE Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/files/move_batch_v2 \
  --header "Authorization: Bearer <get access token>" \
  --header "Content-Type: application/json" \
  --data "{\"entries\": [{\"from_path\": \"/Homework/math\", \"to_path\": \"/Homework/algebra\"}], \"autorename\": false}
```

PARAMETERS

```
{
  "entries": [
    {
      "from_path": "/Homework/math",
      "to_path": "/Homework/algebra"
    }
  ],
  "autorename": false,
  "allow_ownership_transfer": false
}
```

MoveBatchArg

entries *List of (RelocationPath, min_items=1)* List of entries to be moved or copied. Each entry is RelocationPath.

autorename *Boolean* If there's a conflict with any file, have the Dropbox server try to autorename that file to avoid the conflict. The default for this field is False.

allow_ownership_transfer *Boolean* Allow moves by owner even if it would result in an ownership transfer for the content being moved. This does not apply to copies. The default for this field is False.

RETURNS Example: complete

```
{
  ".tag": "complete",
  "entries": [
    {
      ".tag": "success",
      "success": {
        ".tag": "file",
        "name": "Prime_Numbers.txt",
        "id": "id:a4ayc_80_0EAAAAAAAAXw",
        "client_modified": "2015-05-12T15:50:38Z",
        "server_modified": "2015-05-12T15:50:38Z",
        "rev": "alc10ce0dd78",
        "size": 7212,
        "path_lower": "/homework/math/prime_numbers.txt",
        "path_display": "/Homework/math/Prime_Numbers.txt",
        "sharing_info": {
          "read_only": true,
          "parent_shared_folder_id": "84528192421",
          "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"
        },
        "is_downloadable": true,
        "property_groups": [
          {
            "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",
            "fields": [
              {
                "name": "Security Policy",
                "value": "Confidential"
              }
            ]
          }
        ]
      }
    }
  ]
}
```



!7ae41e4649b934ca495991b7852b855 ,

```
        "file_lock_info": {
            "is_lockholder": true,
            "lockholder_name": "Imaginary User",
            "created": "2015-05-12T15:50:38Z"
        }
    }
}
```

Example: `async_job_id`

```
{
    ".tag": "async_job_id",
    "async_job_id": "34g93hh34h04y384084"
}
```

`RelocationBatchV2Launch (union)`

Result returned by `copy_batch:2` or `move_batch:2` that may either launch an asynchronous job or complete synchronously. The value will be one of the following datatypes:

async_job_id `String(min_length=1)` This response indicates that the processing is asynchronous. The string is an id that can be used to obtain the status of the asynchronous job.

complete `RelocationBatchV2Result`

ERRORS No errors.

/move_batch/check

VERSION

2 ▾

DESCRIPTION Returns the status of an asynchronous job for `move_batch:2`. It returns list of results for each entry.

URL STRUCTURE

```
https://api.dropboxapi.com/2/files/move_batch/check_v2
```

AUTHENTICATION [User Authentication, Dropbox-API-Select-Admin \(Team Admin\)](#)

ENDPOINT FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/move_batch/check_v2 \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"async_job_id\": \"34g93hh34h04y384084\"}"
```

PARAMETERS

```
{
    "async_job_id": "34g93hh34h04y384084"
}
```

PollArg

Arguments for methods that poll the status of an asynchronous job. This datatype comes from an imported namespace originally defined in the `async` namespace.

async_job_id `String(min_length=1)` Id of the asynchronous job. This is the value of a response returned from the method that launched the job.

RETURNS

```
{
    ".tag": "complete",
    "entries": [
        {
            ".tag": "success",
            "success": {
                ".tag": "file",
                "name": "Prime_Numbers.txt",
                "id": "id:aayc_80_0EAAAAAAAAXw",
                "client_modified": "2015-05-12T15:50:38Z",
                "server_modified": "2015-05-12T15:50:38Z",
                "rev": "alc10ce0dd78",
                "size": 7212,
                "path_lower": "/homework/math/prime_numbers.txt",
                "path_display": "/Homework/math/Prime_Numbers.txt",
                "sharing_info": {
                    "read_only": true,
                    "parent_shared_folder_id": "84528192421",
                    "modified_by": "dbid:AAH4f99T0ta0NIB-OurWxbNQ6ywGRopQngc"
                },
                "is_downloadable": true,
                "property_groups": [
                    {
                        ".tag": "group"
                    }
                ]
            }
        }
    ]
}
```



Ya",



```
        "value": "Confidential"
    }
}
],
"has_explicit_shared_members": false,
"content_hash": "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",
"file_lock_info": {
    "is_lockholder": true,
    "lockholder_name": "Imaginary User",
    "created": "2015-05-12T15:50:38Z"
}
}
]
```

Example: `in_progress`

```
{
    ".tag": "in_progress"
}
```

`RelocationBatchV2JobStatus (union)`

Result returned by `copy_batch/check:2` or `move_batch/check:2` that may either be in progress or completed with result for each entry. The value will be one of the following datatypes:

in_progress `Void` The asynchronous job is still in progress.

complete `RelocationBatchV2Result` The copy or move batch job has finished.

ERRORS Example: `invalid_async_job_id`

```
{
    "error_summary": "invalid_async_job_id/...",
    "error": {
        ".tag": "invalid_async_job_id"
    }
}
```

Example: `internal_error`

```
{
    "error_summary": "internal_error/...",
    "error": {
        ".tag": "internal_error"
    }
}
```

Example: `other`

```
{
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

`PollError (open union)`

Error returned by methods for polling the status of asynchronous job. This datatype comes from an imported namespace originally defined in the `async` namespace. The value will be one of the following datatypes. New values may be introduced as our API evolves.

invalid_async_job_id `Void` The job ID is invalid.

internal_error `Void` Something went wrong with the job on Dropbox's end. You'll need to verify that the action you were taking succeeded, and if not, try again. This should happen very rarely.

/permanently_delete

VERSION

1 ▾

DESCRIPTION Permanently delete the file or folder at a given path (see <https://www.dropbox.com/en/help/40>).
Note: This endpoint is only available for Dropbox Business apps.

URL STRUCTURE

https://api.dropboxapi.com/2/files/permanently_delete

AUTHENTICATION [User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT
FORMAT

RPC



PARAMETERS

```
{  
    "path": "/Homework/math/Prime_Numbers.txt"  
}
```

DeleteArg

path *String(pattern="/(.|[r|n])*|(ns:[0-9]+(.*?))|(id:*)")* Path in the user's Dropbox to delete.

parent_rev *String(min_length=9, pattern="[0-9a-f]+")?* Perform delete if given "rev" matches the existing file's latest "rev". This field does not support deleting a folder. This field is optional.

RETURNS No return values.

ERRORS Example: too_many_write_operations

```
{  
    "error_summary": "too_many_write_operations/...",  
    "error": {  
        ".tag": "too_many_write_operations"  
    }  
}
```

Example: too_many_files

```
{  
    "error_summary": "too_many_files/...",  
    "error": {  
        ".tag": "too_many_files"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

DeleteError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path_lookup *LookupError*

path_write *WriteError*

too_many_write_operations *Void* There are too many write operations in user's Dropbox. Please retry this request.

too_many_files *Void* There are too many files in one request. Please retry with fewer files.

/restore

VERSION

1

DESCRIPTION

Restore a specific revision of a file to the given path.

URL STRUCTURE

<https://api.dropboxapi.com/2/files/restore>

AUTHENTICATION [User Authentication, Dropbox-API-Select-Admin \(Team Admin\)](#)

ENDPOINT FORMAT

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/restore \  
    --header "Authorization: Bearer <get access token>" \  
    --header "Content-Type: application/json" \  
    --data "{\"path\": \"/root/word.docx\", \"rev\": \"a1c10ce0dd78\"}"
```

PARAMETERS

```
{  
    "path": "/root/word.docx",  
    "rev": "a1c10ce0dd78"  
}
```



rev *String(min_length=9, pattern="[0-9a-f]+")* The revision to restore.

RETURNS

```
{  
    "name": "Prime_Numbers.txt",  
    "id": "id:a4ayc_80_0EAAAAAAAAXw",  
    "client_modified": "2015-05-12T15:50:38Z",  
    "server_modified": "2015-05-12T15:50:38Z",  
    "rev": "alc10ce0dd78",  
    "size": 7212,  
    "path_lower": "/homework/math/prime_numbers.txt",  
    "path_display": "/Homework/math/Prime_Numbers.txt",  
    "sharing_info": {  
        "read_only": true,  
        "parent_shared_folder_id": "84528192421",  
        "modified_by": "dbid:AAH4f99T0ta0NIb-OurWxbNQ6ywGRopQngc"  
    },  
    "is_downloadable": true,  
    "property_groups": [  
        {  
            "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",  
            "fields": [  
                {  
                    "name": "Security Policy",  
                    "value": "Confidential"  
                }  
            ]  
        }  
    ],  
    "has_explicit_shared_members": false,  
    "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",  
    "file_lock_info": {  
        "is_lockholder": true,  
        "lockholder_name": "Imaginary User",  
        "created": "2015-05-12T15:50:38Z"  
    }  
}
```

FileMetadata

name *String* The last component of the path (including extension). This never contains a slash.

id *String(min_length=1)* A unique identifier for the file.

client_modified *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")*

For files, this is the modification time set by the desktop client when the file was added to Dropbox. Since this time is not verified (the Dropbox server stores whatever the desktop client sends up), this should only be used for display purposes (such as sorting) and not, for example, to determine if a file has changed or not.

server_modified *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")*

The last time the file was modified on Dropbox.

rev *String(min_length=9, pattern="[0-9a-f]+")* A unique identifier for the current revision of a file. This field is the same rev as elsewhere in the API and can be used to detect changes and avoid conflicts.

size *UInt64* The file size in bytes.

path_lower *String?* The lowercased full path in the user's Dropbox. This always starts with a slash. This field will be null if the file or folder is not mounted. This field is optional.

path_display *String?* The cased path to be used for display purposes only. In rare instances the casing will not correctly match the user's filesystem, but this behavior will match the path provided in the Core API v1, and at least the last path component will have the correct casing. Changes to only the casing of paths won't be returned by [list_folder/continue](#). This field will be null if the file or folder is not mounted. This field is optional.

parent_shared_folder_id *String(pattern="[-_0-9a-zA-Z:]")?* Field is deprecated. Please use [FileSharingInfo.parent_shared_folder_id](#) or [FolderSharingInfo.parent_shared_folder_id](#) instead. This field is optional.

media_info *MediaInfo?* Additional information if the file is a photo or video. This field will not be set on entries returned by [list_folder](#), [list_folder/continue](#), or [get_thumbnail_batch](#), starting December 2, 2019. This field is optional.

symlink_info *SymlinkInfo?* Set if this file is a symlink. This field is optional.

sharing_info *FileSharingInfo?* Set if this file is contained in a shared folder. This field is optional.

is_downloadable *Boolean* If true, file can be downloaded directly; else the file must be exported. The default for this field is True.

export_info *ExportInfo?* Information about format this file can be exported to. This field must be set if *is_downloadable* is set to false. This field is optional.



has_explicit_shared_members Boolean? This flag will only be present if include_has_explicit_shared_members is true in [list_folder](#) or [get_metadata](#). If this flag is present, it will be true if this file has any explicit shared members. This is different from sharing_info in that this could be true in the case where a file has explicit members but is not contained within a shared folder. This field is optional.

content_hash String(min_length=64, max_length=64)? A hash of the file content. This field can be used to verify data integrity. For more information see our [Content hash](#) page. This field is optional.

file_lock_info FileLockMetadata? If present, the metadata associated with the file's current lock. This field is optional.

ERRORS Example: invalid_revision

```
{  
    "error_summary": "invalid_revision/...",  
    "error": {  
        ".tag": "invalid_revision"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

RestoreError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path_lookup LookupError An error occurs when downloading metadata for the file.

path_write WriteError An error occurs when trying to restore the file to that path.

invalid_revision Void The revision is invalid. It may not exist.

/save_url

VERSION

1 ▾

DESCRIPTION

Save the data from a specified URL into a file in user's Dropbox.

Note that the transfer from the URL must complete within 5 minutes, or the operation will time out and the job will fail.

If the given path already exists, the file will be renamed to avoid the conflict (e.g. myfile (1).txt).

URL STRUCTURE

https://api.dropboxapi.com/2/files/save_url

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/save_url \  
    --header "Authorization: Bearer <get access token>" \  
    --header "Content-Type: application/json" \  
    --data "{\"path\": \"/a.txt\", \"url\": \"http://example.com/a.txt\"}"
```

PARAMETERS

```
{  
    "path": "/a.txt",  
    "url": "http://example.com/a.txt"  
}
```

SaveUrlArg

path String(pattern="/(.|[r\n])*") The path in Dropbox where the URL will be saved to.

url String The URL to be saved.

RETURNS

```
{  
    ".tag": "complete",  
    "name": "Prime_Numbers.txt",  
    "id": "id:a4ayc_80_0EAAAAAAAXw",  
    "client_modified": "2015-05-12T15:50:38Z",  
    "server_modified": "2015-05-12T15:50:38Z",  
}
```



≡

```
        "sharing_info": {
            "read_only": true,
            "parent_shared_folder_id": "84528192421",
            "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"
        },
        "is_downloadable": true,
        "property_groups": [
            {
                "template_id": "ptid:1a5n2i6d30YEAAAAAAAAYa",
                "fields": [
                    {
                        "name": "Security Policy",
                        "value": "Confidential"
                    }
                ]
            }
        ],
        "has_explicit_shared_members": false,
        "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",
        "file_lock_info": {
            "is_lockholder": true,
            "lockholder_name": "Imaginary User",
            "created": "2015-05-12T15:50:38Z"
        }
    }
}
```

SaveUrlResult (*union*)

The value will be one of the following datatypes:

async_job_id *String(min_length=1)* This response indicates that the processing is asynchronous. The string is an id that can be used to obtain the status of the asynchronous job.

complete *FileMetadata* Metadata of the file where the URL is saved to.

ERRORS Example: download_failed

```
{
    "error_summary": "download_failed/...",
    "error": {
        ".tag": "download_failed"
    }
}
```

Example: invalid_url

```
{
    "error_summary": "invalid_url/...",
    "error": {
        ".tag": "invalid_url"
    }
}
```

Example: not_found

```
{
    "error_summary": "not_found/...",
    "error": {
        ".tag": "not_found"
    }
}
```

Example: other

```
{
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

SaveUrlError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path *WriteError*

download_failed *Void* Failed downloading the given URL. The URL may be password-protected and the password provided was incorrect, or the link may be disabled.

invalid_url *Void* The given URL is invalid.

not_found *Void* The file where the URL is saved to no longer exists.

/save_url/check_job_status

VERSION

1

DESCRIPTION Check the status of a [save_url](#) job.



AUTHENTICATION

User Authentication

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/save_url/check_job_status \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"async_job_id\": \"34g93hh34h04y384084\"}"
```

PARAMETERS

```
{  
    "async_job_id": "34g93hh34h04y384084"  
}
```

PollArg

Arguments for methods that poll the status of an asynchronous job. This datatype comes from an imported namespace originally defined in the `async` namespace.

async_job_id *String(min_length=1)* Id of the asynchronous job. This is the value of a response returned from the method that launched the job.

RETURNS

```
{  
    ".tag": "in_progress"  
}
```

SaveUrlJobStatus (*union*)

The value will be one of the following datatypes:

in_progress *Void* The asynchronous job is still in progress.

complete *FileMetadata* Metadata of the file where the URL is saved to.

failed *SaveUrlError*

ERRORS

Example: invalid_async_job_id

```
{  
    "error_summary": "invalid_async_job_id/...",  
    "error": {  
        ".tag": "invalid_async_job_id"  
    }  
}
```

Example: internal_error

```
{  
    "error_summary": "internal_error/...",  
    "error": {  
        ".tag": "internal_error"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

PollError (*open union*)

Error returned by methods for polling the status of asynchronous job. This datatype comes from an imported namespace originally defined in the `async` namespace. The value will be one of the following datatypes. New values may be introduced as our API evolves.

invalid_async_job_id *Void* The job ID is invalid.

internal_error *Void* Something went wrong with the job on Dropbox's end. You'll need to verify that the action you were taking succeeded, and if not, try again. This should happen very rarely.

/search

VERSION

2 ▾

DESCRIPTION

Searches for files and folders.

Note: `search:2` along with `search/continue:2` can only be used to retrieve a maximum of 10,000 matches.



URL STRUCTURE

```
https://api.dropboxapi.com/2/files/search_v2
```

AUTHENTICATION

User Authentication

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/search_v2 \
    --header "Authorization: Bearer <get access token>" \
    --header "Content-Type: application/json" \
    --data "{\"query\": \"cat\", \"include_highlights\": false}"
```

PARAMETERS

```
{  
    "query": "cat",  
    "include_highlights": false  
}
```

SearchV2Arg

query *String* The string to search for. May match across multiple fields based on the request arguments.

options *SearchOptions?* Options for more targeted search results. This field is optional.

include_highlights *Boolean* The default for this field is False.

RETURNS

```
{  
    "matches": [  
        {  
            "metadata": {  
                ".tag": "metadata",  
                "metadata": {  
                    ".tag": "file",  
                    "name": "Prime_Numbers.txt",  
                    "id": "id:a4ayc_80_0EAAAAAAAAXw",  
                    "client_modified": "2015-05-12T15:50:38Z",  
                    "server_modified": "2015-05-12T15:50:38Z",  
                    "rev": "alc10ce0dd78",  
                    "size": 7212,  
                    "path_lower": "/homework/math/prime_numbers.txt",  
                    "path_display": "/Homework/math/Prime_Numbers.txt",  
                    "sharing_info": {  
                        "read_only": true,  
                        "parent_shared_folder_id": "84528192421",  
                        "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"  
                    },  
                    "is_downloadable": true,  
                    "property_groups": [  
                        {  
                            "template_id": "ptid:la5n2i6d30YEAFFFFFFAYa",  
                            "fields": [  
                                {  
                                    "name": "Security Policy",  
                                    "value": "Confidential"  
                                }  
                            ]  
                        }  
                    ],  
                    "has_explicit_shared_members": false,  
                    "content_hash": "e3b0c4429fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",  
                    "file_lock_info": {  
                        "is_lockholder": true,  
                        "lockholder_name": "Imaginary User",  
                        "created": "2015-05-12T15:50:38Z"  
                    }  
                }  
            }  
        }  
    ],  
    "has_more": false  
}
```

SearchV2Result

matches *List of (SearchMatchV2)* A list (possibly empty) of matches for the query.

has_more *Boolean* Used for paging. If true, indicates there is another page of results available that can be fetched by calling `search/continue:2` with the cursor.

cursor *String(min_length=1)?* Pass the cursor into `search/continue:2` to fetch the next page of results. This field is optional.

ERRORS

SearchError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path LookupError

invalid_argument *String?* This field is optional.



Version: 2 ▾

DESCRIPTION Fetches the next page of search results returned from [search:2](#).
Note: [search:2](#) along with [search/continue:2](#) can only be used to retrieve a maximum of 10,000 matches.
Recent changes may not immediately be reflected in search results due to a short delay in indexing. Duplicate results may be returned across pages. Some results may not be returned.

URL STRUCTURE https://api.dropboxapi.com/2/files/search/continue_v2

AUTHENTICATION [User Authentication](#)

ENDPOINT FORMAT [RPC](#)

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/search/continue_v2 \
    --header "Authorization: Bearer <get access token>" \
    --header "Content-Type: application/json" \
    --data "{\"cursor\": \"ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu\"}"
```

PARAMETERS {
 "cursor": "ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu"
}

SearchV2ContinueArg

cursor *String(min_length=1)* The cursor returned by your last call to [search:2](#). Used to fetch the next page of results.

RETURNS {
 "matches": [
 {
 "metadata": {
 ".tag": "metadata",
 "metadata": {
 ".tag": "file",
 "name": "Prime_Numbers.txt",
 "id": "id:a4ayc_80_0EAAAAAAAAXw",
 "client_modified": "2015-05-12T15:50:38Z",
 "server_modified": "2015-05-12T15:50:38Z",
 "rev": "alc10ce0dd78",
 "size": 7212,
 "path_lower": "/homework/math/prime_numbers.txt",
 "path_display": "/Homework/math/Prime_Numbers.txt",
 "sharing_info": {
 "read_only": true,
 "parent_shared_folder_id": "84528192421",
 "modified_by": "dbid:AAH4f99T0taONib-OurWxbNQ6ywGRopQngc"
 },
 "is_downloadable": true,
 "property_groups": [
 {
 "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",
 "fields": [
 {
 "name": "Security Policy",
 "value": "Confidential"
 }
]
 }
],
 "has_explicit_shared_members": false,
 "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",
 "file_lock_info": {
 "is_lockholder": true,
 "lockholder_name": "Imaginary User",
 "created": "2015-05-12T15:50:38Z"
 }
 }
 }
],
 "has_more": false
}

SearchV2Result

matches *List of (SearchMatchV2)* A list (possibly empty) of matches for the query.

has_more *Boolean* Used for paging. If true, indicates there is another page of results available that can be fetched by calling [search/continue:2](#) with the cursor.

cursor *String(min_length=1)?* Pass the cursor into [search/continue:2](#) to fetch the next page of results. This field is optional.

ERRORS [SearchError \(open union\)](#)



y be

**invalid_argument** *String?* This field is optional.

/unlock_file_batch

VERSION

1

PREVIEW - may change or disappear without notice

DESCRIPTION Unlock the files at the given paths. A locked file can only be unlocked by the lock holder or, if a business account, a team admin. A successful response indicates that the file has been unlocked. Returns a list of the unlocked file paths and their metadata after this operation.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

`https://api.dropboxapi.com/2/files/unlock_file_batch`AUTHENTICATION [User Authentication, Dropbox-API-Select-Admin \(Whole Team\)](#)ENDPOINT FORMAT
RPC

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/unlock_file_batch \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"entries\": [{\"path\": \"/John Doe/sample/test.pdf\"}]}"
```

PARAMETERS

```
{
  "entries": [
    {
      "path": "/John Doe/sample/test.pdf"
    }
  ]
}
```

UnlockFileBatchArg

entries *List of (UnlockFileArg)* List of 'entries'. Each 'entry' contains a path of the file which will be unlocked. Duplicate path arguments in the batch are considered only once.

RETURNS

```
{
  "entries": [
    {
      ".tag": "success",
      "metadata": {
        ".tag": "file",
        "name": "Prime_Numbers.txt",
        "id": "id:a4ayc_80_0EAAAAAAAAXw",
        "client_modified": "2015-05-12T15:50:38Z",
        "server_modified": "2015-05-12T15:50:38Z",
        "rev": "alc10ce0dd78",
        "size": 7212,
        "path_lower": "/homework/math/prime_numbers.txt",
        "path_display": "/Homework/math/Prime_Numbers.txt",
        "sharing_info": {
          "read_only": true,
          "parent_shared_folder_id": "84528192421",
          "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"
        },
        "is_downloadable": true,
        "property_groups": [
          {
            "template_id": "ptid:la5n2i6d30YEAFFFFFFAYa",
            "fields": [
              {
                "name": "Security Policy",
                "value": "Confidential"
              }
            ]
          }
        ],
        "has_explicit_shared_members": false,
        "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",
        "file_lock_info": {
          "is_lockholder": true,
          "lockholder_name": "Imaginary User",
          "created": "2015-05-12T15:50:38Z"
        },
        "lock": {
          "content": {
            ".tag": "single_user",
            "created": "2015-05-12T15:50:38Z",
            "lock_holder_account_id": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc",
            "lock_holder_team_id": "dbtid:1234abcd"
          }
        }
      }
    }
  ]
}
```



≡



LOCKFILEBATCHRESULT

entries *List of (LockFileResultEntry)* Each Entry in the 'entries' will have '.tag' with the operation status (e.g. success), the metadata for the file and the lock state after the operation.

ERRORS

Example: too_many_write_operations

```
{  
    "error_summary": "too_many_write_operations/...",  
    "error": {  
        ".tag": "too_many_write_operations"  
    }  
}
```

Example: too_many_files

```
{  
    "error_summary": "too_many_files/...",  
    "error": {  
        ".tag": "too_many_files"  
    }  
}
```

Example: no_write_permission

```
{  
    "error_summary": "no_write_permission/...",  
    "error": {  
        ".tag": "no_write_permission"  
    }  
}
```

Example: cannot_be_locked

```
{  
    "error_summary": "cannot_be_locked/...",  
    "error": {  
        ".tag": "cannot_be_locked"  
    }  
}
```

Example: file_not_shared

```
{  
    "error_summary": "file_not_shared/...",  
    "error": {  
        ".tag": "file_not_shared"  
    }  
}
```

Example: internal_error

```
{  
    "error_summary": "internal_error/...",  
    "error": {  
        ".tag": "internal_error"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

LockFileError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path_lookup *LookupError* Could not find the specified resource.

too_many_write_operations *Void* There are too many write operations in user's Dropbox. Please retry this request.

too_many_files *Void* There are too many files in one request. Please retry with fewer files.

no_write_permission *Void* The user does not have permissions to change the lock state or access the file.

cannot_be_locked *Void* Item is a type that cannot be locked.

file_not_shared *Void* Requested file is not currently shared.

lock_conflict *LockConflictError* The user action conflicts with an existing lock on the file.

internal_error *Void* Something went wrong with the job on Dropbox's end. You'll need to verify that the action you were taking succeeded, and if not, try again. This should happen very rarely.



Version: 1 ▾

DESCRIPTION Create a new file with the contents provided in the request.
Do not use this to upload a file larger than 150 MB. Instead, create an upload session with [upload_session/start](#).
Calls to this endpoint will count as data transport calls for any Dropbox Business teams with a limit on the number of data transport calls allowed per month. For more information, see the [Data transport limit page](#).

URL STRUCTURE <https://content.dropboxapi.com/2/files/upload>

AUTHENTICATION [User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT FORMAT Content-upload

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://content.dropboxapi.com/2/files/upload \
  -header "Authorization: Bearer <get access token>" \
  -header "Dropbox-API-Arg: {\"path\": \"/Homework/math/Matrices.txt\", \"mode\": \"add\", \"autorename\": true, \""
  -header "Content-Type: application/octet-stream" \
  -data-binary @local_file.txt
```

PARAMETERS

```
{\n    \"path\": \"/Homework/math/Matrices.txt\",\\n    \"mode\": \"add\",\\n    \"autorename\": true,\\n    \"mute\": false,\\n    \"strict_conflict\": false\\n}
```

Example: update

```
{\n    \"path\": \"/Homework/math/Matrices.txt\",\\n    \"mode\": {\n        \".tag\": \"update\",\\n        \"update\": \"alc10ce0dd78\"\n    },\\n    \"autorename\": false,\\n    \"mute\": false,\\n    \"property_groups\": [\n        {\n            \"template_id\": \"ptid:la5n2i6d30YEAAAAAAAAYa\",\\n            \"fields\": [\n                {\n                    \"name\": \"Security Policy\",\\n                    \"value\": \"Confidential\"\n                }\n            ]\n        }\n    ],\\n    \"strict_conflict\": false\n}
```

CommitInfo

path *String(pattern="(.|[\\r\\n])*|(ns:[0-9]+(.*?))|(id:*)")* Path in the user's Dropbox to save the file.

mode *WriteMode* Selects what to do if the file already exists. The default for this union is add.

autorename *Boolean* If there's a conflict, as determined by mode, have the Dropbox server try to autorename the file to avoid conflict. The default for this field is False.

client_modified *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")?* The value to store as the client_modified timestamp. Dropbox automatically records the time at which the file was written to the Dropbox servers. It can also record an additional timestamp, provided by Dropbox desktop clients, mobile clients, and API apps of when the file was actually created or modified. This field is optional.

mute *Boolean* Normally, users are made aware of any file modifications in their Dropbox account via notifications in the client software. If true, this tells the clients that this modification shouldn't result in a user notification. The default for this field is False.

property_groups *List of (PropertyGroup)?* List of custom properties to add to file. This field is optional.

strict_conflict *Boolean* Be more strict about how each WriteMode detects conflict. For example, always return a conflict error when mode = WriteMode.update and the given "rev" doesn't match the existing file's "rev", even if the existing file has been deleted. The default for this field is False.

RETURNS



≡

```
  "client_modified": "2015-05-12T15:50:38Z",
  "server_modified": "2015-05-12T15:50:38Z",
  "rev": "a1c10ce0dd78",
  "size": 7212,
  "path_lower": "/homework/math/prime_numbers.txt",
  "path_display": "/Homework/math/Prime_Numbers.txt",
  "sharing_info": {
    "read_only": true,
    "parent_shared_folder_id": "84528192421",
    "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"
  },
  "is_downloadable": true,
  "property_groups": [
    {
      "template_id": "ptid:1a5n2i6d30YEAAAAAAAAYa",
      "fields": [
        {
          "name": "Security Policy",
          "value": "Confidential"
        }
      ]
    }
  ],
  "has_explicit_shared_members": false,
  "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",
  "file_lock_info": {
    "is_lockholder": true,
    "lockholder_name": "Imaginary User",
    "created": "2015-05-12T15:50:38Z"
  }
}
```

FileMetadata

name *String* The last component of the path (including extension). This never contains a slash.

id *String(min_length=1)* A unique identifier for the file.

client_modified *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")*
For files, this is the modification time set by the desktop client when the file was added to Dropbox. Since this time is not verified (the Dropbox server stores whatever the desktop client sends up), this should only be used for display purposes (such as sorting) and not, for example, to determine if a file has changed or not.

server_modified *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")*
The last time the file was modified on Dropbox.

rev *String(min_length=9, pattern="[0-9a-f]+")* A unique identifier for the current revision of a file. This field is the same rev as elsewhere in the API and can be used to detect changes and avoid conflicts.

size *UInt64* The file size in bytes.

path_lower *String?* The lowercased full path in the user's Dropbox. This always starts with a slash. This field will be null if the file or folder is not mounted. This field is optional.

path_display *String?* The cased path to be used for display purposes only. In rare instances the casing will not correctly match the user's filesystem, but this behavior will match the path provided in the Core API v1, and at least the last path component will have the correct casing. Changes to only the casing of paths won't be returned by [list_folder/continue](#). This field will be null if the file or folder is not mounted. This field is optional.

parent_shared_folder_id (deprecated) *String(pattern="[-_0-9a-zA-Z:]")?* Field is deprecated. Please use FileSharingInfo.parent_shared_folder_id or FolderSharingInfo.parent_shared_folder_id instead. This field is optional.

media_info *MediaInfo?* Additional information if the file is a photo or video. This field will not be set on entries returned by [list_folder](#), [list_folder/continue](#), or [get_thumbnail_batch](#), starting December 2, 2019. This field is optional.

symlink_info *SymlinkInfo?* Set if this file is a symlink. This field is optional.

sharing_info *FileSharingInfo?* Set if this file is contained in a shared folder. This field is optional.

is_downloadable *Boolean* If true, file can be downloaded directly; else the file must be exported. The default for this field is True.

export_info *ExportInfo?* Information about format this file can be exported to. This field must be set if **is_downloadable** is set to false. This field is optional.

property_groups *List of (PropertyGroup)?* Additional information if the file has custom properties with the property template specified. This field is optional.

has_explicit_shared_members *Boolean?* This flag will only be present if include_has_explicit_shared_members is true in [list_folder](#) or [get_metadata](#). If this flag is present, it will be true if this file has any explicit shared members. This is different from sharing_info in



the file content. This field can be used to verify data integrity. For more information see our [Content hash](#) page. This field is optional.

file_lock_info *FileLockMetadata?* If present, the metadata associated with the file's current lock. This field is optional.

ERRORS

UploadError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path *UploadWriteFailed* Unable to save the uploaded contents to a file.

properties_error *InvalidPropertyGroupError* The supplied property group is invalid. The file has uploaded without property groups.

/upload_session/append

VERSION

2

DESCRIPTION

Append more data to an upload session.

When the parameter close is set, this call will close the session.

A single request should not upload more than 150 MB. The maximum size of a file one can upload to an upload session is 350 GB.

Calls to this endpoint will count as data transport calls for any Dropbox Business teams with a limit on the number of data transport calls allowed per month. For more information, see the [Data transport limit page](#).

URL STRUCTURE

https://content.dropboxapi.com/2/files/upload_session/append_v2

AUTHENTICATION

[User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT FORMAT

[Content-upload](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://content.dropboxapi.com/2/files/upload_session/append_v2 \
  --header "Authorization: Bearer <get access token>" \
  --header "Dropbox-API-Arg: {\"cursor\": {\"session_id\": \"1234faaf0678bcde\", \"offset\": 0}, \"close\": false}" \
  --header "Content-Type: application/octet-stream" \
  --data-binary @local_file.txt
```

PARAMETERS

```
{
  "cursor": {
    "session_id": "1234faaf0678bcde",
    "offset": 0
  },
  "close": false
}
```

UploadSessionAppendArg

cursor *UploadSessionCursor* Contains the upload session ID and the offset.

close *Boolean* If true, the current session will be closed, at which point you won't be able to call [upload_session/append:2](#) anymore with the current session. The default for this field is False.

RETURNS

No return values.

ERRORS

Example: not_found

```
{
  "error_summary": "not_found/...",
  "error": {
    ".tag": "not_found"
  }
}
```

Example: closed

```
{
  "error_summary": "closed/...",
  "error": {
    ".tag": "closed"
  }
}
```

Example: not_closed



≡

```
        ".tag": "not_closed"
    }
```

Example: too_large

```
{
    "error_summary": "too_large/...",
    "error": {
        ".tag": "too_large"
    }
}
```

Example: other

```
{
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

UploadSessionLookupError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

not_found *Void* The upload session ID was not found or has expired. Upload sessions are valid for 48 hours.

incorrect_offset *UploadSessionOffsetError* The specified offset was incorrect. See the value for the correct offset. This error may occur when a previous request was received and processed successfully but the client did not receive the response, e.g. due to a network error.

closed *Void* You are attempting to append data to an upload session that has already been closed (i.e. committed).

not_closed *Void* The session must be closed before calling `upload_session/finish_batch`.

too_large *Void* You can not append to the upload session because the size of a file should not reach the max file size limit (i.e. 350GB).

/upload_session/finish

VERSION

1 ▾

DESCRIPTION

Finish an upload session and save the uploaded data to the given file path.

A single request should not upload more than 150 MB. The maximum size of a file one can upload to an upload session is 350 GB.

Calls to this endpoint will count as data transport calls for any Dropbox Business teams with a limit on the number of data transport calls allowed per month. For more information, see the [Data transport limit page](#).

URL STRUCTURE

https://content.dropboxapi.com/2/files/upload_session/finish

AUTHENTICATION

[User Authentication, Dropbox-API-Select-Admin \(Team Admin\)](#)

ENDPOINT
FORMAT

[Content-upload](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://content.dropboxapi.com/2/files/upload_session/finish \
--header "Authorization: Bearer <get access token>" \
--header "Dropbox-API-Arg: {\"cursor\": {\"session_id\": \"1234faaf0678bcde\", \"offset\": 0}, \"commit\": {\"path\": \"/Homework/math/Matrices.txt\", \"mode\": \"add\", \"autorename\": true, \"mute\": false, \"strict_conflict\": false}}
```

PARAMETERS

```
{
    "cursor": {
        "session_id": "1234faaf0678bcde",
        "offset": 0
    },
    "commit": {
        "path": "/Homework/math/Matrices.txt",
        "mode": "add",
        "autorename": true,
        "mute": false,
        "strict_conflict": false
    }
}
```

Example: update



```
        "offset": 0
    },
    "commit": {
        "path": "/Homework/math/Matrices.txt",
        "mode": {
            ".tag": "update",
            "update": "a1c10ce0dd78"
        },
        "autorename": false,
        "mute": false,
        "property_groups": [
            {
                "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",
                "fields": [
                    {
                        "name": "Security Policy",
                        "value": "Confidential"
                    }
                ]
            }
        ],
        "strict_conflict": false
    }
}
```

UploadSessionFinishArg

cursor *UploadSessionCursor* Contains the upload session ID and the offset.

commit *CommitInfo* Contains the path and other optional modifiers for the commit.

RETURNS

```
{
    "name": "Prime_Numbers.txt",
    "id": "id:a4ayc_80_OEAAAAAAAAXw",
    "client_modified": "2015-05-12T15:50:38Z",
    "server_modified": "2015-05-12T15:50:38Z",
    "rev": "a1c10ce0dd78",
    "size": 7212,
    "path_lower": "/homework/math/prime_numbers.txt",
    "path_display": "/Homework/math/Prime_Numbers.txt",
    "sharing_info": {
        "read_only": true,
        "parent_shared_folder_id": "84528192421",
        "modified_by": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"
    },
    "is_downloadable": true,
    "property_groups": [
        {
            "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",
            "fields": [
                {
                    "name": "Security Policy",
                    "value": "Confidential"
                }
            ]
        }
    ],
    "has_explicit_shared_members": false,
    "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",
    "file_lock_info": {
        "is_lockholder": true,
        "lockholder_name": "Imaginary User",
        "created": "2015-05-12T15:50:38Z"
    }
}
```

FileMetadata

name *String* The last component of the path (including extension). This never contains a slash.

id *String(min_length=1)* A unique identifier for the file.

client_modified *Timestamp(format="%Y-%m-%dT%H:%M:%S")* For files, this is the modification time set by the desktop client when the file was added to Dropbox. Since this time is not verified (the Dropbox server stores whatever the desktop client sends up), this should only be used for display purposes (such as sorting) and not, for example, to determine if a file has changed or not.

server_modified *Timestamp(format="%Y-%m-%dT%H:%M:%S")* The last time the file was modified on Dropbox.

rev *String(min_length=9, pattern="[0-9a-f]+")* A unique identifier for the current revision of a file. This field is the same rev as elsewhere in the API and can be used to detect changes and avoid conflicts.

size *UInt64* The file size in bytes.

path_lower *String?* The lowercased full path in the user's Dropbox. This always starts with a slash. This field will be null if the file or folder is not mounted. This field is optional.

path_display *String?* The cased path to be used for display purposes only. In rare instances the casing will not correctly match the user's filesystem, but this behavior will match the path provided in the Core API v1, and at least the last path component will have the correct casing. Changes to only the casing of paths won't be



Z:+")? Field is deprecated. Please use `FileSharingInfo.parent_shared_folder_id` or `FolderSharingInfo.parent_shared_folder_id` instead. This field is optional.

media_info `MediaInfo?` Additional information if the file is a photo or video. This field will not be set on entries returned by `list_folder`, `list_folder/continue`, or `get_thumbnail_batch`, starting December 2, 2019. This field is optional.

symlink_info `SymlinkInfo?` Set if this file is a symlink. This field is optional.

sharing_info `FileSharingInfo?` Set if this file is contained in a shared folder. This field is optional.

is_downloadable `Boolean` If true, file can be downloaded directly; else the file must be exported. The default for this field is True.

export_info `ExportInfo?` Information about format this file can be exported to. This field must be set if `is_downloadable` is set to false. This field is optional.

property_groups `List of (PropertyGroup)?` Additional information if the file has custom properties with the property template specified. This field is optional.

has_explicit_shared_members `Boolean?` This flag will only be present if `include_has_explicit_shared_members` is true in `list_folder` or `get_metadata`. If this flag is present, it will be true if this file has any explicit shared members. This is different from `sharing_info` in that this could be true in the case where a file has explicit members but is not contained within a shared folder. This field is optional.

content_hash `String(min_length=64, max_length=64)?` A hash of the file content. This field can be used to verify data integrity. For more information see our [Content hash](#) page. This field is optional.

file_lock_info `FileLockMetadata?` If present, the metadata associated with the file's current lock. This field is optional.

ERRORS Example: too_many_shared_folder_targets

```
{  
    "error_summary": "too_many_shared_folder_targets/...",  
    "error": {  
        ".tag": "too_many_shared_folder_targets"  
    }  
}
```

Example: too_many_write_operations

```
{  
    "error_summary": "too_many_write_operations/...",  
    "error": {  
        ".tag": "too_many_write_operations"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

UploadSessionFinishError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

lookup_failed `UploadSessionLookupError` The session arguments are incorrect; the value explains the reason.

path `WriteError` Unable to save the uploaded contents to a file. Data has already been appended to the upload session. Please retry with empty data body and updated offset.

properties_error `InvalidPropertyGroupError` The supplied property group is invalid. The file has uploaded without property groups.

too_many_shared_folder_targets (deprecated) `Void` Field is deprecated. The batch request commits files into too many different shared folders. Please limit your batch request to files contained in a single shared folder.

too_many_write_operations `Void` There are too many write operations happening in the user's Dropbox. You should retry uploading this file.



DESCRIPTION This route helps you commit many files at once into a user's Dropbox. Use [upload_session/start](#) and [upload_session/append:2](#) to upload file contents. We recommend uploading many files in parallel to increase throughput. Once the file contents have been uploaded, rather than calling [upload_session/finish](#), use this route to finish all your upload sessions in a single request.

UploadSessionStartArg.close or UploadSessionAppendArg.close needs to be true for the last [upload_session/start](#) or [upload_session/append:2](#) call. The maximum size of a file one can upload to an upload session is 350 GB.

This route will return a job_id immediately and do the async commit job in background. Use [upload_session/finish_batch/check](#) to check the job status.

For the same account, this route should be executed serially. That means you should not start the next job before current job finishes. We allow up to 1000 entries in a single request. Calls to this endpoint will count as data transport calls for any Dropbox Business teams with a limit on the number of data transport calls allowed per month. For more information, see the [Data transport limit page](#).

URL STRUCTURE

```
https://api.dropboxapi.com/2/files/upload_session/finish_batch
```

AUTHENTICATION

User Authentication, Dropbox-API-Select-Admin (Team Admin)

ENDPOINT FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/upload_session/finish_batch \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"entries\": [{\"cursor\": {\"session_id\": \"1234faaf0678bcde\"}, \"offset\": 0}, {\"commit\": {\"path\": \"/Homework/math/Matrices.txt\", \"mode\": \"add\", \"autorename\": true, \"mute\": false, \"strict_conflict\": false}}]}
```

PARAMETERS

```
{
  "entries": [
    {
      "cursor": {
        "session_id": "1234faaf0678bcde",
        "offset": 0
      },
      "commit": {
        "path": "/Homework/math/Matrices.txt",
        "mode": "add",
        "autorename": true,
        "mute": false,
        "strict_conflict": false
      }
    }
  ]
}
```

UploadSessionFinishBatchArg

entries List of ([UploadSessionFinishArg](#), max_items=1000)

Commit information for each file in the batch.

RETURNS

Example: complete

```
{
  ".tag": "complete",
  "entries": [
    {
      ".tag": "success",
      "name": "Prime_Numbers.txt",
      "id": "id:a4ayc_80_0EAAAAAAAAXw",
      "client_modified": "2015-05-12T15:50:38Z",
      "server_modified": "2015-05-12T15:50:38Z",
      "rev": "alc10ce0dd78",
      "size": 7212,
      "path_lower": "/homework/math/prime_numbers.txt",
      "path_display": "/Homework/math/Prime_Numbers.txt",
      "sharing_info": {
        "read_only": true,
        "parent_shared_folder_id": "84528192421",
        "modified_by": "dbid:AAH4f99T0ta0NIb-OurWxbNQ6ywGRopQngc"
      },
      "is_downloadable": true,
      "property_groups": [
        {
          "template_id": "ptid:la5n2i6d30YEAFFFFFFAYa",
          "fields": [
            {
              "name": "Security Policy",
              "value": "Confidential"
            }
          ]
        }
      ],
      "has_explicit_shared_members": false,
      "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",
      "file_lock_info": {
        "is_lockholder": true,
        "lockholder_name": "Imaginary User",
        "lockholder_id": "dbid:AAH4f99T0ta0NIb-OurWxbNQ6ywGRopQngc"
      }
    }
  ]
}
```



≡

}

Example: `async_job_id`{
 `".tag": "async_job_id",`
 `"async_job_id": "34g93hh34h04y384084"`
}Example: `other`{
 `".tag": "other"`
}`UploadSessionFinishBatchLaunch` (*open union*)Result returned by `upload_session/finish_batch` that may either launch an asynchronous job or complete synchronously. The value will be one of the following datatypes. New values may be introduced as our API evolves.**async_job_id** `String(min_length=1)` This response indicates that the processing is asynchronous. The string is an id that can be used to obtain the status of the asynchronous job.**complete** `UploadSessionFinishBatchResult`

ERRORS No errors.

[/upload_session/finish_batch/check](#)

VERSION

1

DESCRIPTION Returns the status of an asynchronous job for `upload_session/finish_batch`. If success, it returns list of result for each entry.

URL STRUCTURE

https://api.dropboxapi.com/2/files/upload_session/finish_batch/checkAUTHENTICATION [User Authentication, Dropbox-API-Select-Admin \(Team Admin\)](#)

ENDPOINT FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/files/upload_session/finish_batch/check \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"async_job_id\": \"34g93hh34h04y384084\"}"
```

PARAMETERS

{
 `"async_job_id": "34g93hh34h04y384084"`
}

PollArg

Arguments for methods that poll the status of an asynchronous job. This datatype comes from an imported namespace originally defined in the `async` namespace.**async_job_id** `String(min_length=1)` Id of the asynchronous job. This is the value of a response returned from the method that launched the job.

RETURNS

{
 `".tag": "complete",`
 `"entries": [`
 {
 `".tag": "success",`
 `"name": "Prime_Numbers.txt",`
 `"id": "id:a4ayc_80_0EAAAAAAAAXw",`
 `"client_modified": "2015-05-12T15:50:38Z",`
 `"server_modified": "2015-05-12T15:50:38Z",`
 `"rev": "alc10ce0dd78",`
 `"size": 7212,`
 `"path_lower": "/homework/math/prime_numbers.txt",`
 `"path_display": "/Homework/math/Prime_Numbers.txt",`
 `"sharing_info": {`
 `"read_only": true,`
 `"parent_shared_folder_id": "84528192421",`
 `"modified_by": "dbid:AAH4f99T0ta0NIb-OurWxbNQ6ywGRopQngc"`
 },
 `"is_downloadable": true,`
 `"property_groups": [`
 {
 `"template_id": "ptid:1a5n2i6d30YEAFFFFFFAYa",`
 }
]
 }
]



≡

```
        }
    ]
},
"has_explicit_shared_members": false,
"content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",
"file_lock_info": {
    "is_lockholder": true,
    "lockholder_name": "Imaginary User",
    "created": "2015-05-12T15:50:38Z"
}
]
```

Example: in_progress

```
{
    ".tag": "in_progress"
}
```

UploadSessionFinishBatchJobStatus (*union*)

The value will be one of the following datatypes:

in_progress *Void* The asynchronous job is still in progress.

complete *UploadSessionFinishBatchResult* The [upload_session/finish_batch](#) has finished.

ERRORS Example: invalid_async_job_id

```
{
    "error_summary": "invalid_async_job_id/...",
    "error": {
        ".tag": "invalid_async_job_id"
    }
}
```

Example: internal_error

```
{
    "error_summary": "internal_error/...",
    "error": {
        ".tag": "internal_error"
    }
}
```

Example: other

```
{
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

PollError (*open union*)

Error returned by methods for polling the status of asynchronous job. This datatype comes from an imported namespace originally defined in the async namespace. The value will be one of the following datatypes. New values may be introduced as our API evolves.

invalid_async_job_id *Void* The job ID is invalid.

internal_error *Void* Something went wrong with the job on Dropbox's end. You'll need to verify that the action you were taking succeeded, and if not, try again. This should happen very rarely.

/upload_session/start

VERSION

1 ▾

DESCRIPTION

Upload sessions allow you to upload a single file in one or more requests, for example where the size of the file is greater than 150 MB. This call starts a new upload session with the given data. You can then use [upload_session/append:2](#) to add more data and [upload_session/finish](#) to save all the data to a file in Dropbox.

A single request should not upload more than 150 MB. The maximum size of a file one can upload to an upload session is 350 GB.

An upload session can be used for a maximum of 48 hours. Attempting to use an `UploadSessionStartResult.session_id` with [upload_session/append:2](#) or [upload_session/finish](#) more than 48 hours after its creation will return a `UploadSessionLookupError.not_found`.

Calls to this endpoint will count as data transport calls for any Dropbox Business teams with a limit on the number of data transport calls allowed per month. For more information, see the [Data transport limit page](#).



AUTHENTICATION

User Authentication, Dropbox-API-Select-Admin (Team Admin)

ENDPOINT
FORMAT

Content-upload

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://content.dropboxapi.com/2/files/upload_session/start \
--header "Authorization: Bearer <get access token>" \
--header "Dropbox-API-Arg: {\"close\": false}" \
--header "Content-Type: application/octet-stream" \
--data-binary @local_file.txt
```

PARAMETERS

```
{\n    \"close\": false\n}
```

UploadSessionStartArg

close Boolean If true, the current session will be closed, at which point you won't be able to call [upload_session/append:2](#) anymore with the current session. The default for this field is False.

RETURNS

```
{\n    \"session_id\": \"1234faaf0678bcde\"\n}
```

UploadSessionStartResult

session_id String A unique identifier for the upload session. Pass this to [upload_session/append:2](#) and [upload_session/finish](#).

ERRORS

No errors.

paper

This namespace contains endpoints and data types for managing docs and folders in Dropbox Paper. New Paper users will see docs they create in their filesystem as '.paper' files alongside their other Dropbox content. The /paper endpoints are being deprecated and you'll need to use /files and /sharing endpoints to interact with their Paper content. Read more in the [Paper Migration Guide](#).

sharing

This namespace contains endpoints and data types for creating and managing shared links and shared folders.

/add_file_member

VERSION

1 ▾

DESCRIPTION

Adds specified members to a file.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

```
https://api.dropboxapi.com/2/sharing/add_file_member
```

AUTHENTICATION

User Authentication, Dropbox-API-Select-Admin (Team Admin)

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/add_file_member \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"file\": \"id:3kmLmQFnf1AAAAAAAAAAw\", \"members\": [{\"tag\": \"email\", \"email\": \"justin@example.com\"}], \"custom_message\": \"This is a custom message about ACME.doc\", \"quiet\": false, \"access_level\": \"viewer\", \"allow_activities\": false}">
```

PARAMETERS

```
{\n    \"file\": \"id:3kmLmQFnf1AAAAAAAAAAw\", \n    \"members\": [\n        {\n            \".tag\": \"email\", \n            \"email\": \"justin@example.com\"\n        }\n    ], \n    \"custom_message\": \"This is a custom message about ACME.doc\", \n    \"quiet\": false, \n    \"access_level\": \"viewer\", \n    \"allow_activities\": false\n}
```



Aaafilememberargs

Arguments for [add_file_member](#).

file *String(min_length=1, pattern="((/id:).*|nspath:[0-9]+:*)|ns:[0-9]+(.*)?")* File to which to add members.

members *List of (MemberSelector)* Members to add. Note that even an email address is given, this may result in a user being directly added to the membership if that email is the user's main account email.

custom_message *String?* Message to send to added members in their invitation. This field is optional.

quiet *Boolean* Whether added members should be notified via device notifications of their invitation. The default for this field is False.

access_level *AccessLevel* AccessLevel union object, describing what access level we want to give new members. The default for this union is viewer.

add_message_as_comment *Boolean* If the custom message should be added as a comment on the file. The default for this field is False.

RETURNS This route returns a list. This means the route can accept a homogenous list of the following types:

FileMemberActionResult

Per-member result for [add_file_member](#) or [change_file_member_access](#).

member *MemberSelector* One of specified input members.

result *FileMemberActionIndividualResult* The outcome of the action on this member.

ERRORS Example: rate_limit

```
{  
    "error_summary": "rate_limit/...",  
    "error": {  
        ".tag": "rate_limit"  
    }  
}
```

Example: invalid_comment

```
{  
    "error_summary": "invalid_comment/...",  
    "error": {  
        ".tag": "invalid_comment"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

AddFileMemberError (*open union*)

Errors for [add_file_member](#). The value will be one of the following datatypes. New values may be introduced as our API evolves.

user_error *SharingUserError*

access_error *SharingFileAccessError*

rate_limit *Void* The user has reached the rate limit for invitations.

invalid_comment *Void* The custom message did not pass comment permissions checks.

/add_folder_member

VERSION

1

DESCRIPTION Allows an owner or editor (if the ACL update policy allows) of a shared folder to add another member.

For the new member to get access to all the functionality for this folder, you will need to call [mount_folder](#) on their behalf.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/add_folder_member



FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/add_folder_member \
  -header "Authorization: Bearer <get access token>" \
  -header "Content-Type: application/json" \
  --data "{\"shared_folder_id\": \"84528192421\", \"members\": [{\"member\": {\".tag\": \"email\", \"email\": \"justin@example.com\"}, \"access_level\": \"editor\"}, {\"member\": {\".tag\": \"dropbox_id\", \"dropbox_id\": \"dbid:AAEufNrMPSPe0dMQijRP0N_aZtBJRm26W4Q\"}, \"access_level\": \"viewer\"}], \"quiet\": false, \"custom_message\": \"Documentation for launch day\"}
```

PARAMETERS

```
{\n    \"shared_folder_id\": \"84528192421\", \n    \"members\": [\n        {\n            \"member\": {\n                \".tag\": \"email\", \n                \"email\": \"justin@example.com\"}, \n            \"access_level\": \"editor\"}, \n            {\n                \"member\": {\n                    \".tag\": \"dropbox_id\", \n                    \"dropbox_id\": \"dbid:AAEufNrMPSPe0dMQijRP0N_aZtBJRm26W4Q\"}, \n                    \"access_level\": \"viewer\"}\n        ], \n        \"quiet\": false, \n        \"custom_message\": \"Documentation for launch day\"}\n}
```

AddFolderMemberArg

shared_folder_id *String(pattern="[-_0-9a-zA-Z:]")* The ID for the shared folder.

members *List of (AddMember)* The intended list of members to add. Added members will receive invites to join the shared folder.

quiet *Boolean* Whether added members should be notified via email and device notifications of their invite. The default for this field is False.

custom_message *String(min_length=1)?* Optional message to display to added members in their invitation. This field is optional.

RETURNS

No return values.

ERRORS

```
{\n    \"error_summary\": \"no_permission/...\", \n    \"error\": {\n        \".tag\": \"no_permission\"\n    }\n}
```

Example: member

```
{\n    \"error_summary\": \"bad_member/invalid_dropbox_id/...\", \n    \"error\": {\n        \".tag\": \"bad_member\", \n        \"bad_member\": {\n            \".tag\": \"invalid_dropbox_id\", \n            \"invalid_dropbox_id\": \"dbid:AAEufNrMPSPe0dMQijRP0N_aZtBJRm26W4Q\"}\n    }\n}
```

Example: email_unverified

```
{\n    \"error_summary\": \"email_unverified/...\", \n    \"error\": {\n        \".tag\": \"email_unverified\"\n    }\n}
```

Example: banned_member

```
{\n    \"error_summary\": \"banned_member/...\", \n    \"error\": {\n        \".tag\": \"banned_member\"\n    }\n}
```

Example: cant_share_outside_team

```
{\n    \"error_summary\": \"cant_share_outside_team/...\", \n    \"error\": {\n        \".tag\": \"cant_share_outside_team\"\n    }\n}
```



≡



Example: rate_limit

```
{  
    "error_summary": "rate_limit/...",  
    "error": {  
        ".tag": "rate_limit"  
    }  
}
```

Example: too_many_invitees

```
{  
    "error_summary": "too_many_invitees/...",  
    "error": {  
        ".tag": "too_many_invitees"  
    }  
}
```

Example: insufficient_plan

```
{  
    "error_summary": "insufficient_plan/...",  
    "error": {  
        ".tag": "insufficient_plan"  
    }  
}
```

Example: team_folder

```
{  
    "error_summary": "team_folder/...",  
    "error": {  
        ".tag": "team_folder"  
    }  
}
```

Example: no_permission

```
{  
    "error_summary": "no_permission/...",  
    "error": {  
        ".tag": "no_permission"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

AddFolderMemberError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

access_error *SharedFolderAccessError* Unable to access shared folder.

email_unverified *Void* The current user's e-mail address is unverified.

banned_member *Void* The current user has been banned.

bad_member *AddMemberSelectorError*

AddFolderMemberArg.members contains a bad invitation recipient.

cant_share_outside_team *Void* Your team policy does not allow sharing outside of the team.

too_many_members *UInt64* The value is the member limit that was reached.

too_many_pending_invites *UInt64* The value is the pending invite limit that was reached.

rate_limit *Void* The current user has hit the limit of invites they can send per day. Try again in 24 hours.

too_many_invitees *Void* The current user is trying to share with too many people at once.

insufficient_plan *Void* The current user's account doesn't support this action. An example of this is when adding a read-only member. This action can only be performed by users that have upgraded to a Pro or Business plan.

team_folder *Void* This action cannot be performed on a team shared folder.

no_permission *Void* The current user does not have permission to perform this action.



Version: 1 ▾

DESCRIPTION Returns the status of an asynchronous job.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

`https://api.dropboxapi.com/2/sharing/check_job_status`

AUTHENTICATION [User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/check_job_status \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"async_job_id\": \"34g93hh34h04y384084\"}"
```

PARAMETERS

```
{\n    \"async_job_id\": \"34g93hh34h04y384084\"\n}
```

PollArg

Arguments for methods that poll the status of an asynchronous job. This datatype comes from an imported namespace originally defined in the `async` namespace.

async_job_id *String(min_length=1)* Id of the asynchronous job.

This is the value of a response returned from the method that launched the job.

RETURNS

Example: `in_progress`

```
{\n    \".tag\": \"in_progress\"\n}
```

Example: `complete`

```
{\n    \".tag\": \"complete\"\n}
```

JobStatus (*union*)

The value will be one of the following datatypes:

in_progress *Void* The asynchronous job is still in progress.

complete *Void* The asynchronous job has finished.

failed *JobError* The asynchronous job returned an error.

ERRORS

Example: `invalid_async_job_id`

```
{\n    \"error_summary\": \"invalid_async_job_id/...\", \n    \"error\": {\n        \".tag\": \"invalid_async_job_id\"\n    }\n}
```

Example: `internal_error`

```
{\n    \"error_summary\": \"internal_error/...\", \n    \"error\": {\n        \".tag\": \"internal_error\"\n    }\n}
```

Example: `other`

```
{\n    \"error_summary\": \"other/...\", \n    \"error\": {\n        \".tag\": \"other\"\n    }\n}
```

PollError (*open union*)

Error returned by methods for polling the status of asynchronous job. This datatype comes from an imported namespace originally defined in the `async` namespace. The value will be one of the



internal_error *Void* Something went wrong with the job on Dropbox's end. You'll need to verify that the action you were taking succeeded, and if not, try again. This should happen very rarely.

/check_remove_member_job_status

VERSION

1

DESCRIPTION Returns the status of an asynchronous job for sharing a folder.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/check_remove_member_job_status

AUTHENTICATION [User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT FORMAT
RPC

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/check_remove_member_job_status \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"async_job_id\": \"34g93hh34h04y384084\"}"
```

PARAMETERS

```
{  
    "async_job_id": "34g93hh34h04y384084"  
}
```

PollArg

Arguments for methods that poll the status of an asynchronous job. This datatype comes from an imported namespace originally defined in the `async` namespace.

async_job_id *String(min_length=1)* Id of the asynchronous job. This is the value of a response returned from the method that launched the job.

RETURNS

```
{  
    ".tag": "complete"  
}
```

Example: in_progress

```
{  
    ".tag": "in_progress"  
}
```

RemoveMemberJobStatus (*union*)

The value will be one of the following datatypes:

in_progress *Void* The asynchronous job is still in progress.

complete *MemberAccessLevelResult* Removing the folder member has finished. The value is information about whether the member has another form of access.

failed *RemoveFolderMemberError*

ERRORS Example: invalid_async_job_id

```
{  
    "error_summary": "invalid_async_job_id/...",  
    "error": {  
        ".tag": "invalid_async_job_id"  
    }  
}
```

Example: internal_error

```
{  
    "error_summary": "internal_error/...",  
    "error": {  
        ".tag": "internal_error"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```



PollError (*open union*)

Error returned by methods for polling the status of asynchronous job. This datatype comes from an imported namespace originally defined in the `async` namespace. The value will be one of the following datatypes. New values may be introduced as our API evolves.

invalid_async_job_id *Void* The job ID is invalid.

internal_error *Void* Something went wrong with the job on Dropbox's end. You'll need to verify that the action you were taking succeeded, and if not, try again. This should happen very rarely.

/check_share_job_status

VERSION

1

DESCRIPTION Returns the status of an asynchronous job for sharing a folder.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/check_share_job_status

AUTHENTICATION [User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/sharing/check_share_job_status \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"async_job_id\": \"34g93hh34h04y384084\"}"
```

PARAMETERS

```
{  
    "async_job_id": "34g93hh34h04y384084"  
}
```

PollArg

Arguments for methods that poll the status of an asynchronous job. This datatype comes from an imported namespace originally defined in the `async` namespace.

async_job_id *String(min_length=1)* Id of the asynchronous job.

This is the value of a response returned from the method that launched the job.

RETURNS

```
{  
    ".tag": "complete",  
    "access_type": {  
        ".tag": "owner"  
    },  
    "is_inside_team_folder": false,  
    "is_team_folder": false,  
    "name": "dir",  
    "policy": {  
        "acl_update_policy": {  
            ".tag": "owner"  
        },  
        "shared_link_policy": {  
            ".tag": "anyone"  
        },  
        "member_policy": {  
            ".tag": "anyone"  
        },  
        "resolved_member_policy": {  
            ".tag": "team"  
        }  
    },  
    "preview_url": "https://www.dropbox.com/scl/fo/fir9vjelf",  
    "shared_folder_id": "84528192421",  
    "time_invited": "2016-01-20T00:00:00Z",  
    "path_lower": "/dir",  
    "link_metadata": {  
        "audience_options": [  
            {  
                ".tag": "public"  
            },  
            {  
                ".tag": "team"  
            },  
            {  
                ".tag": "members"  
            }  
        ],  
        "current_audience": {  
            ".tag": "public"  
        },  
    },  
}
```



≡

```
        },
        "allow": true
    }
],
"password_protected": false,
"url": ""
},
"permissions": [],
"access_inheritance": {
    ".tag": "inherit"
}
}
```

Example: `in_progress`

```
{
    ".tag": "in_progress"
}
```

`ShareFolderJobStatus (union)`

The value will be one of the following datatypes:

in_progress `Void` The asynchronous job is still in progress.

complete `SharedFolderMetadata` The share job has finished. The value is the metadata for the folder.

failed `ShareFolderError`

ERRORS Example: `invalid_async_job_id`

```
{
    "error_summary": "invalid_async_job_id/...",
    "error": {
        ".tag": "invalid_async_job_id"
    }
}
```

Example: `internal_error`

```
{
    "error_summary": "internal_error/...",
    "error": {
        ".tag": "internal_error"
    }
}
```

Example: `other`

```
{
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

`PollError (open union)`

Error returned by methods for polling the status of asynchronous job. This datatype comes from an imported namespace originally defined in the `async` namespace. The value will be one of the following datatypes. New values may be introduced as our API evolves.

invalid_async_job_id `Void` The job ID is invalid.

internal_error `Void` Something went wrong with the job on Dropbox's end. You'll need to verify that the action you were taking succeeded, and if not, try again. This should happen very rarely.

/create_shared_link_with_settings

VERSION

1 ▾

DESCRIPTION Create a shared link with custom settings. If no settings are given then the default visibility is `RequestedVisibility.public` (The resolved visibility, though, may depend on other aspects such as team and shared folder settings).

URL STRUCTURE

`https://api.dropboxapi.com/2/sharing/create_shared_link_with_settings`

AUTHENTICATION [User Authentication, Dropbox-API-Select-Admin \(Team Admin\)](#)

ENDPOINT FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/create_shared_link_with_settings \
--header "Authorization: Bearer <get access token> \" \\"
```



PARAMETERS

```
{  
    "path": "/Prime_Numbers.txt",  
    "settings": {  
        "requested_visibility": "public",  
        "audience": "public",  
        "access": "editor"  
    }  
}
```

CreateSharedLinkWithSettingsArg

path String(pattern="/(.|[\\r\\n])*[id:.*](rev:[0-9a-f]{9,})(ns:[0-9]+)(.*)?") The path to be shared by the shared link.

settings SharedLinkSettings? The requested settings for the newly created shared link. This field is optional.

RETURNS

```
{  
    ".tag": "file",  
    "url": "https://www.dropbox.com/s/2sn712vy1ovegw8/Prime_Numbers.txt?dl=0",  
    "name": "Prime_Numbers.txt",  
    "link_permissions": {  
        "can_revoke": false,  
        "resolved_visibility": {  
            ".tag": "public"  
        },  
        "revoke_failure_reason": {  
            ".tag": "owner_only"  
        }  
    },  
    "client_modified": "2015-05-12T15:50:38Z",  
    "server_modified": "2015-05-12T15:50:38Z",  
    "rev": "a1c10ce0dd78",  
    "size": 7212,  
    "id": "id:a4ayc_80_0EAAAAAAAAXw",  
    "path_lower": "/homework/math/prime_numbers.txt",  
    "team_member_info": {  
        "team_info": {  
            "id": "dbtid:AAFdgehTzw7WlXhZJsbGCLePe8RvQGYDr-I",  
            "name": "Acme, Inc."  
        },  
        "display_name": "Roger Rabbit",  
        "member_id": "dbmid:abcd1234"  
    }  
}
```

```
{  
    ".tag": "folder",  
    "url": "https://www.dropbox.com/sh/s6fvw6ol7rmqolx/AAAgWRShjYDvPpDB30Sykjfa?dl=0",  
    "name": "Math",  
    "link_permissions": {  
        "can_revoke": false,  
        "resolved_visibility": {  
            ".tag": "public"  
        },  
        "revoke_failure_reason": {  
            ".tag": "owner_only"  
        }  
    },  
    "id": "id:a4ayc_80_0EAAAAAAAAXw",  
    "path_lower": "/homework/math",  
    "team_member_info": {  
        "team_info": {  
            "id": "dbtid:AAFdgehTzw7WlXhZJsbGCLePe8RvQGYDr-I",  
            "name": "Acme, Inc."  
        },  
        "display_name": "Roger Rabbit",  
        "member_id": "dbmid:abcd1234"  
    }  
}
```

SharedLinkMetadata (*datatype with subtypes*)

The metadata of a shared link. This datatype will be one of the following subtypes:

file FileLinkMetadata

folder FolderLinkMetadata

ERRORS Example: email_not_verified

```
{  
    "error_summary": "email_not_verified/...",  
    "error": {  
        ".tag": "email_not_verified"  
    }  
}
```

Example: access_denied

```
{  
    "error_summary": "access_denied/...",  
    "error": {  
        ".tag": "access_denied"  
    }  
}
```



email_not_verified *Void* User's email should be verified.

shared_link_already_exists *SharedLinkAlreadyExistsMetadata?* The shared link already exists. You can call [list_shared_links](#) to get the existing link, or use the provided metadata if it is returned. This field is optional.

settings_error *SharedLinkSettingsError* There is an error with the given settings.

access_denied *Void* Access to the requested path is forbidden.

/get_file_metadata

VERSION

1

DESCRIPTION Returns shared file metadata.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/get_file_metadata

AUTHENTICATION

[User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/sharing/get_file_metadata \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"file\": {\"id:3kmLmQFnf1AAAAAAAAAAw\", \"actions\": []}}
```

PARAMETERS

```
{  
    "file": "id:3kmLmQFnf1AAAAAAAAAAw",  
    "actions": []  
}
```

GetFileMetadataArg

Arguments of [get_file_metadata](#).

file *String(min_length=1, pattern="((/id:).*)|nspath:[0-9]+(:*)|ns:[0-9]+(:*)?")* The file to query.

actions *List of (FileAction)?* A list of `FileAction`'s corresponding to `FilePermission`'s that should appear in the response's SharedFileMetadata.permissions field describing the actions the authenticated user can perform on the file. This field is optional.

RETURNS

```
{  
    "id": "id:3kmLmQFnf1AAAAAAAAAAw",  
    "name": "file.txt",  
    "policy": {  
        "acl_update_policy": {  
            ".tag": "owner"  
        },  
        "shared_link_policy": {  
            ".tag": "anyone"  
        },  
        "member_policy": {  
            ".tag": "anyone"  
        },  
        "resolved_member_policy": {  
            ".tag": "team"  
        }  
    },  
    "preview_url": "https://www.dropbox.com/scl/fi/fir9vjelf",  
    "access_type": {  
        ".tag": "viewer"  
    },  
    "owner_display_names": [  
        "Jane Doe"  
    ],  
    "owner_team": {  
        "id": "dbtid:AAFdgehTzw7WlXhZJsbGCLePe8RvQGYDr-I",  
        "name": "Acme, Inc."  
    },  
    "path_display": "/dir/file.txt",  
    "path_lower": "/dir/file.txt",  
    "permissions": [],  
    "time_invited": "2016-01-20T00:00:00Z"  
}
```

SharedFileMetadata

Properties of the shared file.

id *String(min_length=4, pattern="id:.+")* The ID of the file.

name *String* The name of this file.



access_type *AccessLevel?* The current user's access level for this shared file. This field is optional.

expected_link_metadata *ExpectedSharedContentLinkMetadata?* The expected metadata of the link associated for the file when it is first shared. Absent if the link already exists. This is for an unreleased feature so it may not be returned yet. This field is optional.

link_metadata *SharedContentLinkMetadata?* The metadata of the link associated for the file. This is for an unreleased feature so it may not be returned yet. This field is optional.

owner_display_names *List of (String)?* The display names of the users that own the file. If the file is part of a team folder, the display names of the team admins are also included. Absent if the owner display names cannot be fetched. This field is optional.

owner_team *Team?* The team that owns the file. This field is not present if the file is not owned by a team. This field is optional.

parent_shared_folder_id *String(pattern="[-_0-9a-zA-Z:]")?* The ID of the parent shared folder. This field is present only if the file is contained within a shared folder. This field is optional.

path_display *String?* The cased path to be used for display purposes only. In rare instances the casing will not correctly match the user's filesystem, but this behavior will match the path provided in the Core API v1. Absent for unmounted files. This field is optional.

path_lower *String?* The lower-case full path of this file. Absent for unmounted files. This field is optional.

permissions *List of (FilePermission)?* The sharing permissions that requesting user has on this file. This corresponds to the entries given in GetFileMetadataBatchArg.actions or GetFileMetadataArg.actions. This field is optional.

time_invited *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")?* Timestamp indicating when the current user was invited to this shared file. If the user was not invited to the shared file, the timestamp will indicate when the user was invited to the parent shared folder. This value may be absent. This field is optional.

ERRORS

GetFileMetadataError (*open union*)

Error result for [get_file_metadata](#). The value will be one of the following datatypes. New values may be introduced as our API evolves.

user_error *SharingUserError*

access_error *Sharing FileAccess Error*

/get_file_metadata/batch

VERSION

1 ▾

DESCRIPTION

Returns shared file metadata.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/get_file_metadata/batch

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/get_file_metadata/batch \
    --header "Authorization: Bearer <get access token>" \
    --header "Content-Type: application/json" \
    --data "{\"files\": [\"id:3kmLmQFnf1AAAAAAAAAAw\", \"id:VvTaJu2VzAAAAAAAADQ\"], \"actions\": []}"
```

PARAMETERS

```
{
  "files": [
    "id:3kmLmQFnf1AAAAAAAAAAw",
    "id:VvTaJu2VzAAAAAAAADQ"
  ],
  "actions": []
}
```

[GetFileMetadataBatchArg](#)

Arguments of [get_file_metadata/batch](#).



ng to

`FilePermission`'s that should appear in the response's SharedFileMetadata.permissions field describing the actions the authenticated user can perform on the file. This field is optional.

RETURNS This route returns a list. This means the route can accept a homogenous list of the following types:
GetFileMetadataBatchResult
Per file results of [get_file_metadata/batch](#).
file *String(min_length=1, pattern="((/id:).*)|nspath:[0-9]+::*)|ns:[0-9]+(.*)?"* This is the input file identifier corresponding to one of GetFileMetadataBatchArg.files.
result *GetFileMetadataIndividualResult* The result for this particular file.

ERRORS Example: email_unverified

```
{  
    "error_summary": "email_unverified/...",  
    "error": {  
        ".tag": "email_unverified"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

SharingUserError (*open union*)

User account had a problem preventing this action. The value will be one of the following datatypes. New values may be introduced as our API evolves.

email_unverified *Void* The current user must verify the account e-mail address before performing this action.

/get_folder_metadata

VERSION

1 ▾

DESCRIPTION Returns shared folder metadata by its folder ID.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/get_folder_metadata

AUTHENTICATION [User Authentication, Dropbox-API-Select-Admin \(Whole Team\)](#)

ENDPOINT FORMAT

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/get_folder_metadata \  
    --header "Authorization: Bearer <get access token>" \  
    --header "Content-Type: application/json" \  
    --data "{\"shared_folder_id\": \"84528192421\", \"actions\": []}"
```

PARAMETERS

```
{  
    "shared_folder_id": "84528192421",  
    "actions": []  
}
```

GetMetadataArgs

shared_folder_id *String(pattern="[-_0-9a-zA-Z:]")* The ID for the shared folder.

actions *List of (FolderAction)?* A list of `FolderAction`'s corresponding to `FolderPermission`'s that should appear in the response's SharedFolderMetadata.permissions field describing the actions the authenticated user can perform on the folder. This field is optional.

RETURNS

```
{  
    "access_type": {  
        ".tag": "owner"  
    },  
}
```



```
        "acl_update_policy": {
            ".tag": "owner"
        },
        "shared_link_policy": {
            ".tag": "anyone"
        },
        "member_policy": {
            ".tag": "anyone"
        },
        "resolved_member_policy": {
            ".tag": "team"
        }
    },
    "preview_url": "https://www.dropbox.com/scl/fo/fir9vjelf",
    "shared_folder_id": "84528192421",
    "time_invited": "2016-01-20T00:00:00Z",
    "path_lower": "/dir",
    "link_metadata": {
        "audience_options": [
            {
                ".tag": "public"
            },
            {
                ".tag": "team"
            },
            {
                ".tag": "members"
            }
        ],
        "current_audience": {
            ".tag": "public"
        },
        "link_permissions": [
            {
                "action": {
                    ".tag": "change_audience"
                },
                "allow": true
            }
        ],
        "password_protected": false,
        "url": ""
    },
    "permissions": [],
    "access_inheritance": {
        ".tag": "inherit"
    }
}
```

SharedFolderMetadata

The metadata which includes basic information about the shared folder.

access_type *AccessLevel* The current user's access level for this shared folder.

is_inside_team_folder *Boolean* Whether this folder is inside of a team folder.

is_team_folder *Boolean* Whether this folder is a [team folder](#).

name *String* The name of the this shared folder.

policy *FolderPolicy* Policies governing this shared folder.

preview_url *String* URL for displaying a web preview of the shared folder.

shared_folder_id *String(pattern="[-_0-9a-zA-Z:]")* The ID of the shared folder.

time_invited *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")*

Timestamp indicating when the current user was invited to this shared folder.

owner_display_names *List of (String)?* The display names of the users that own the folder. If the folder is part of a team folder, the display names of the team admins are also included. Absent if the owner display names cannot be fetched. This field is optional.

owner_team *Team?* The team that owns the folder. This field is not present if the folder is not owned by a team. This field is optional.

parent_shared_folder_id *String(pattern="[-_0-9a-zA-Z:]")?* The ID of the parent shared folder. This field is present only if the folder is contained within another shared folder. This field is optional.

path_lower *String?* The lower-cased full path of this shared folder. Absent for unmounted folders. This field is optional.

parent_folder_name *String?* Display name for the parent folder. This field is optional.

link_metadata *SharedContentLinkMetadata?* The metadata of the shared content link to this shared folder. Absent if there is no link on the folder. This is for an unreleased feature so it may not be returned yet. This field is optional.

permissions *List of (FolderPermission)?* Actions the current user may perform on the folder and its contents. The set of permissions corresponds to the FolderActions in the request. This field is optional.



ERRORS Example: invalid_id

```
{  
    "error_summary": "invalid_id/...",  
    "error": {  
        ".tag": "invalid_id"  
    }  
}
```

Example: not_a_member

```
{  
    "error_summary": "not_a_member/...",  
    "error": {  
        ".tag": "not_a_member"  
    }  
}
```

Example: email_unverified

```
{  
    "error_summary": "email_unverified/...",  
    "error": {  
        ".tag": "email_unverified"  
    }  
}
```

Example: unmounted

```
{  
    "error_summary": "unmounted/...",  
    "error": {  
        ".tag": "unmounted"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

SharedFolderAccessError (*open union*)

There is an error accessing the shared folder. The value will be one of the following datatypes. New values may be introduced as our API evolves.

invalid_id *Void* This shared folder ID is invalid.

not_a_member *Void* The user is not a member of the shared folder thus cannot access it.

email_unverified [deprecated] *Void* Field is deprecated. Never set.

unmounted *Void* The shared folder is unmounted.

/get_shared_link_file

VERSION

1

DESCRIPTION Download the shared link's file from a user's Dropbox.

URL STRUCTURE

https://content.dropboxapi.com/2/sharing/get_shared_link_file

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[Content-download](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://content.dropboxapi.com/2/sharing/get_shared_link_file \  
    -header "Authorization: Bearer <get access token>" \  
    -header "Dropbox-API-Arg: {\\"url\\": \\"https://www.dropbox.com/s/2sn712vy1ovegw8/Prime_Numbers.txt?dl=0\\\", \\"pat
```

PARAMETERS

```
{  
    "url": "https://www.dropbox.com/s/2sn712vy1ovegw8/Prime_Numbers.txt?dl=0",  
    "path": "/Prime_Numbers.txt"  
}
```

[GetSharedLinkMetadataArg](#)

url *String* URL of the shared link.



is optional.

link_password *String?* If the shared link has a password, this parameter can be used. This field is optional.

RETURNS

```
{  
    ".tag": "file",  
    "url": "https://www.dropbox.com/s/2sn712vylogegw8/Prime_Numbers.txt?dl=0",  
    "name": "Prime_Numbers.txt",  
    "link_permissions": {  
        "can_revoke": false,  
        "resolved_visibility": {  
            ".tag": "public"  
        },  
        "revoke_failure_reason": {  
            ".tag": "owner_only"  
        }  
    },  
    "client_modified": "2015-05-12T15:50:38Z",  
    "server_modified": "2015-05-12T15:50:38Z",  
    "rev": "a1c10ce0dd78",  
    "size": 7212,  
    "id": "id:a4ayc_80_0EAAAAAAAAXw",  
    "path_lower": "/homework/math/prime_numbers.txt",  
    "team_member_info": {  
        "team_info": {  
            "id": "dbtid:AAFdgehTzw7WlXhZJsbGCLePe8RvQGYDr-I",  
            "name": "Acme, Inc."  
        },  
        "display_name": "Roger Rabbit",  
        "member_id": "dbmid:abcd1234"  
    }  
}
```

```
{  
    ".tag": "folder",  
    "url": "https://www.dropbox.com/sh/s6fvw6o17rmqolx/AAAgWRSbjmYDvPpDB30Sykjfa?dl=0",  
    "name": "Math",  
    "link_permissions": {  
        "can_revoke": false,  
        "resolved_visibility": {  
            ".tag": "public"  
        },  
        "revoke_failure_reason": {  
            ".tag": "owner_only"  
        }  
    },  
    "id": "id:a4ayc_80_0EAAAAAAAAXw",  
    "path_lower": "/homework/math",  
    "team_member_info": {  
        "team_info": {  
            "id": "dbtid:AAFdgehTzw7WlXhZJsbGCLePe8RvQGYDr-I",  
            "name": "Acme, Inc."  
        },  
        "display_name": "Roger Rabbit",  
        "member_id": "dbmid:abcd1234"  
    }  
}
```

SharedLinkMetadata (*datatype with subtypes*)

The metadata of a shared link. This datatype will be one of the following subtypes:

file *FileLinkMetadata*

folder *FolderLinkMetadata*

ERRORS

Example: shared_link_not_found

```
{  
    "error_summary": "shared_link_not_found/...",  
    "error": {  
        ".tag": "shared_link_not_found"  
    }  
}
```

Example: shared_link_access_denied

```
{  
    "error_summary": "shared_link_access_denied/...",  
    "error": {  
        ".tag": "shared_link_access_denied"  
    }  
}
```

Example: unsupported_link_type

```
{  
    "error_summary": "unsupported_link_type/...",  
    "error": {  
        ".tag": "unsupported_link_type"  
    }  
}
```

Example: other



≡

```
        ".tag": "other"
    }
```

Example: shared_link_is_directory

```
{
    "error_summary": "shared_link_is_directory/...",
    "error": {
        ".tag": "shared_link_is_directory"
    }
}
```

GetSharedLinkFileError (*union*)

The value will be one of the following datatypes:

shared_link_not_found *Void* The shared link wasn't found.

shared_link_access_denied *Void* The caller is not allowed to access this shared link.

unsupported_link_type *Void* This type of link is not supported; use [files/export](#) instead.

shared_link_is_directory *Void* Directories cannot be retrieved by this endpoint.

/get_shared_link_metadata

VERSION

1 ▾

DESCRIPTION Get the shared link's metadata.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/get_shared_link_metadata

AUTHENTICATION [User Authentication](#)

ENDPOINT FORMAT [RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/get_shared_link_metadata \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"url\": \"https://www.dropbox.com/s/2sn712vylovegw8/Prime_Numbers.txt?dl=0\", \"path\": \"/Prime_Numbe
```

PARAMETERS

```
{
    "url": "https://www.dropbox.com/s/2sn712vylovegw8/Prime_Numbers.txt?dl=0",
    "path": "/Prime_Numbers.txt"
}
```

GetSharedLinkMetadataArg

url *String* URL of the shared link.

path *String*(*pattern*=*/(.|[r\n])*?*)? If the shared link is to a folder, this parameter can be used to retrieve the metadata for a specific file or sub-folder in this folder. A relative path should be used. This field is optional.

link_password *String*? If the shared link has a password, this parameter can be used. This field is optional.

RETURNS

```
{
    ".tag": "file",
    "url": "https://www.dropbox.com/s/2sn712vylovegw8/Prime_Numbers.txt?dl=0",
    "name": "Prime_Numbers.txt",
    "link_permissions": {
        "can_revoke": false,
        "resolved_visibility": {
            ".tag": "public"
        },
        "revoke_failure_reason": {
            ".tag": "owner_only"
        }
    },
    "client_modified": "2015-05-12T15:50:38Z",
    "server_modified": "2015-05-12T15:50:38Z",
    "rev": "a1c10ce0dd78",
    "size": 7212,
    "id": "id:aayc_80_OEAAAAAAAAXw",
    "path_lower": "/homework/math/prime_numbers.txt",
    "team_member_info": {
        "team_info": {
            "id": "dbtid:AAFdgehTzw7WlXhZJsbGCLePe8RvQGYDr-I",
            "name": "Acme, Inc."
        },
        "display_name": "Roger Rabbit",
        "member_id": "dbmid:abcd1234"
    }
}
```



≡

```
{  
    ".tag": "folder",  
    "url": "https://www.dropbox.com/sh/s6fvw6ol7rmqolx/AAAgWRSbjmYDvPpDB30Sykjfa?dl=0",  
    "name": "Math",  
    "link_permissions": {  
        "can_revoke": false,  
        "resolved_visibility": {  
            ".tag": "public"  
        },  
        "revoke_failure_reason": {  
            ".tag": "owner_only"  
        }  
    },  
    "id": "id:a4ayc_80_OEAAAAAAAAXw",  
    "path_lower": "/homework/math",  
    "team_member_info": {  
        "team_info": {  
            "id": "dbtid:AAFdgehTzw7WlXhZJsbGCLePe8RvQGYDr-I",  
            "name": "Acme, Inc."  
        },  
        "display_name": "Roger Rabbit",  
        "member_id": "dbmid:abcd1234"  
    }  
}
```

SharedLinkMetadata (*datatype with subtypes*)

The metadata of a shared link. This datatype will be one of the following subtypes:

file *FileLinkMetadata*

folder *FolderLinkMetadata*

EXAMPLES: `shared_link_not_found`

```
{  
    "error_summary": "shared_link_not_found/...",  
    "error": {  
        ".tag": "shared_link_not_found"  
    }  
}
```

EXAMPLES: `shared_link_access_denied`

```
{  
    "error_summary": "shared_link_access_denied/...",  
    "error": {  
        ".tag": "shared_link_access_denied"  
    }  
}
```

EXAMPLES: `unsupported_link_type`

```
{  
    "error_summary": "unsupported_link_type/...",  
    "error": {  
        ".tag": "unsupported_link_type"  
    }  
}
```

EXAMPLES: `other`

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

SharedLinkError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

shared_link_not_found *Void* The shared link wasn't found.

shared_link_access_denied *Void* The caller is not allowed to access this shared link.

unsupported_link_type *Void* This type of link is not supported; use [files/export](#) instead.

/list_file_members

VERSION

1 ▾

DESCRIPTION Use to obtain the members who have been invited to a file, both inherited and uninherited members.

This endpoint does not support apps with the app folder permission.



AUTHENTICATION

User Authentication, Dropbox-API-Select-Admin (Team Admin)

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/list_file_members \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"file\": \"id:3kmLmQFnf1AAAAAAAAAAw\", \"include_inherited\": true, \"limit\": 100}"
```

PARAMETERS

```
{  
  "file": "id:3kmLmQFnf1AAAAAAAAAAw",  
  "include_inherited": true,  
  "limit": 100  
}
```

ListFileMembersArg

Arguments for [list_file_members](#).

file *String(min_length=1, pattern="((/|id:).*)|nspath:[0-9]+(:.*))|ns:[0-9]+(/.*)?"* The file for which you want to see members.

actions *List of (MemberAction)?* The actions for which to return permissions on a member. This field is optional.

include_inherited *Boolean* Whether to include members who only have access from a parent shared folder. The default for this field is True.

limit *UInt32(min=1, max=300)* Number of members to return max per query. Defaults to 100 if no limit is specified. The default for this field is 100.

RETURNS

```
{  
  "users": [  
    {  
      "access_type": {  
        ".tag": "owner"  
      },  
      "user": {  
        "account_id": "dbid:AAH4f99T0taONib-OurWxbNQ6ywGRopQngc",  
        "email": "bob@example.com",  
        "display_name": "Robert Smith",  
        "same_team": true,  
        "team_member_id": "dbmid:abcd1234"  
      },  
      "permissions": [],  
      "is_inherited": false,  
      "time_last_seen": "2016-01-20T00:00:00Z",  
      "platform_type": {  
        ".tag": "unknown"  
      }  
    }  
  ],  
  "groups": [  
    {  
      "access_type": {  
        ".tag": "editor"  
      },  
      "group": {  
        "group_name": "Test group",  
        "group_id": "g:e2db7665347abcd60000000001a2b3c",  
        "group_management_type": {  
          ".tag": "user_managed"  
        },  
        "group_type": {  
          ".tag": "user_managed"  
        },  
        "is_member": false,  
        "is_owner": false,  
        "same_team": true,  
        "member_count": 10  
      },  
      "permissions": [],  
      "is_inherited": false  
    }  
  ],  
  "invitees": [  
    {  
      "access_type": {  
        ".tag": "viewer"  
      },  
      "invitee": {  
        ".tag": "email",  
        "email": "jessica@example.com"  
      },  
      "permissions": [],  
      "is_inherited": false  
    }  
  ]  
}
```

SharedFileMembers

Shared file user, group, and invitee membership.
Used for the results of [list_file_members](#) and



ers of

the shared file.

groups *List of (GroupMembershipInfo)* The list of group members of the shared file.**invitees** *List of (InviteeMembershipInfo)* The list of invited members of a file, but have not logged in and claimed this.**cursor** *String?* Present if there are additional shared file members that have not been returned yet. Pass the cursor into **list_file_members/continue** to list additional members. This field is optional.**ERRORS****ListFileMembersError** (*open union*)Error for **list_file_members**. The value will be one of the following datatypes. New values may be introduced as our API evolves.**user_error** *SharingUserError***access_error** *Sharing FileAccess Error*[/list_file_members/batch](#)

VERSION

1



DESCRIPTION

Get members of multiple files at once. The arguments to this route are more limited, and the limit on query result size per file is more strict. To customize the results more, use the individual file endpoint.

Inherited users and groups are not included in the result, and permissions are not returned for this endpoint.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/list_file_members/batch

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/sharing/list_file_members/batch \
  -header "Authorization: Bearer <get access token>" \
  -header "Content-Type: application/json" \
  -data "{\"files\": [\"id:3kmLmQFnf1AAAAAAAAAAw\", \"id:VvTaJu2VZzAAAAAAAADQ\"], \"limit\": 10}"
```

PARAMETERS

```
{\n    \"files\": [\n        \"id:3kmLmQFnf1AAAAAAAAAAw\",\n        \"id:VvTaJu2VZzAAAAAAAADQ\"\n    ],\n    \"limit\": 10\n}
```

[ListFileMembersBatchArg](#)Arguments for [list_file_members/batch](#).**files** *List of (String(min_length=1, pattern="(|id:).*)|nspath:[0-9]+(:*)|ns:[0-9]+(.*)?"), max_items=100)* Files for which to return members.**limit** *UInt32(max=20)* Number of members to return max per query. Defaults to 10 if no limit is specified. The default for this field is 10.

RETURNS

This route returns a list. This means the route can accept a homogenous list of the following types:

[ListFileMembersBatchResult](#)Per-file result for [list_file_members/batch](#).**file** *String(min_length=1, pattern="(|id:).*)|nspath:[0-9]+(:*)|ns:[0-9]+(.*)?")* This is the input file identifier, whether an ID or a path.**result** *ListFileMembersIndividualResult* The result for this particular file.

ERRORS

Example: email_unverified

```
{\n    \"error_summary\": \"email_unverified/...\", \n    \"error\": {\n        \".tag\": \"email_unverified\"\n    }\n}
```



≡

```
"error_summary": "other/...",
"error": {
    ".tag": "other"
}
```

SharingUserError (*open union*)

User account had a problem preventing this action. The value will be one of the following datatypes. New values may be introduced as our API evolves.

email_unverified *Void* The current user must verify the account e-mail address before performing this action.

/list_file_members/continue

VERSION

1

DESCRIPTION Once a cursor has been retrieved from [list_file_members](#) or [list_file_members/batch](#), use this to paginate through all shared file members.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/list_file_members/continue

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/sharing/list_file_members/continue \
      -header "Authorization: Bearer <get access token>" \
      -header "Content-Type: application/json" \
      -data "{\"cursor\": \"ZtkX9_EHj3x7PMkVuFIhwKYXEpwLwyxp9vMKomUhllil9q7eWiAu\"}"
```

PARAMETERS

```
{\n    "cursor": "ZtkX9_EHj3x7PMkVuFIhwKYXEpwLwyxp9vMKomUhllil9q7eWiAu"\n}
```

ListFileMembersContinueArg

Arguments for [list_file_members/continue](#).

cursor *String* The cursor returned by your last call to [list_file_members](#), [list_file_members/continue](#), or [list_file_members/batch](#).

RETURNS

```
{\n    "users": [\n        {\n            "access_type": {\n                ".tag": "owner"\n            },\n            "user": {\n                "account_id": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc",\n                "email": "bob@example.com",\n                "display_name": "Robert Smith",\n                "same_team": true,\n                "team_member_id": "dbmid:abcd1234"\n            },\n            "permissions": [],\n            "is_inherited": false,\n            "time_last_seen": "2016-01-20T00:00:00Z",\n            "platform_type": {\n                ".tag": "unknown"\n            }\n        }\n    ],\n    "groups": [\n        {\n            "access_type": {\n                ".tag": "editor"\n            },\n            "group": {\n                "group_name": "Test group",\n                "group_id": "g:e2db7665347abcd60000000001a2b3c",\n                "group_management_type": {\n                    ".tag": "user_managed"\n                },\n                "group_type": {\n                    ".tag": "user_managed"\n                },\n                "is_member": false,\n                "is_owner": false,\n                "same_team": true,\n                "member_count": 10\n            },\n            "permissions": []\n        }\n    ]\n}
```



≡

```
{  
    "access_type": {  
        ".tag": "viewer"  
    },  
    "invitee": {  
        ".tag": "email",  
        "email": "jessica@example.com"  
    },  
    "permissions": [],  
    "is_inherited": false  
}  
]  
}
```

SharedFileMembers

Shared file user, group, and invitee membership.

Used for the results of [list_file_members](#) and [list_file_members/continue](#), and used as part of the results for [list_file_members/batch](#).

users *List of (UserFileMembershipInfo)* The list of user members of the shared file.

groups *List of (GroupMembershipInfo)* The list of group members of the shared file.

invitees *List of (InviteeMembershipInfo)* The list of invited members of a file, but have not logged in and claimed this.

cursor *String?* Present if there are additional shared file members that have not been returned yet. Pass the cursor into [list_file_members/continue](#) to list additional members. This field is optional.

ERRORS

Example: invalid_cursor

```
{  
    "error_summary": "invalid_cursor/...",  
    "error": {  
        ".tag": "invalid_cursor"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

ListFileMembersContinueError (*open union*)

Error for [list_file_members/continue](#). The value will be one of the following datatypes. New values may be introduced as our API evolves.

user_error *SharingUserError*

access_error *Sharing FileAccess Error*

invalid_cursor *Void* ListFileMembersContinueArg.cursor is invalid.

/list_folder_members

VERSION

1 ▾

DESCRIPTION Returns shared folder membership by its folder ID.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/list_folder_members

AUTHENTICATION [User Authentication, Dropbox-API-Select-Admin \(Whole Team\)](#)

ENDPOINT FORMAT

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/list_folder_members \  
--header "Authorization: Bearer <get access token>" \  
--header "Content-Type: application/json" \  
--data "{\"shared_folder_id\": \"84528192421\", \"actions\": [], \"limit\": 10}"
```

PARAMETERS

```
{  
    "shared_folder_id": "84528192421",
```



≡



ListFolderMembersArgs

shared_folder_id *String(pattern="[-_0-9a-zA-Z:]")* The ID for the shared folder.

actions *List of (MemberAction)?* This is a list indicating whether each returned member will include a boolean value MemberPermission.allow that describes whether the current user can perform the MemberAction on the member. This field is optional.

limit *UInt32(min=1, max=1000)* The maximum number of results that include members, groups and invitees to return per request. The default for this field is 1000.

RETURNS

```
{  
    "users": [  
        {  
            "access_type": {  
                ".tag": "owner"  
            },  
            "user": {  
                "account_id": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc",  
                "email": "bob@example.com",  
                "display_name": "Robert Smith",  
                "same_team": true,  
                "team_member_id": "dbmid:abcd1234"  
            },  
            "permissions": [],  
            "is_inherited": false  
        }  
    ],  
    "groups": [  
        {  
            "access_type": {  
                ".tag": "editor"  
            },  
            "group": {  
                "group_name": "Test group",  
                "group_id": "g:e2db7665347abcd600000000001a2b3c",  
                "group_management_type": {  
                    ".tag": "user_managed"  
                },  
                "group_type": {  
                    ".tag": "user_managed"  
                },  
                "is_member": false,  
                "is_owner": false,  
                "same_team": true,  
                "member_count": 10  
            },  
            "permissions": [],  
            "is_inherited": false  
        }  
    ],  
    "invitees": [  
        {  
            "access_type": {  
                ".tag": "viewer"  
            },  
            "invitee": {  
                ".tag": "email",  
                "email": "jessica@example.com"  
            },  
            "permissions": [],  
            "is_inherited": false  
        }  
    ],  
    "cursor": "ZtkX9_Ehj3x7PMkVuFIhwKYXEwpwLwyxp9vMKomUhllil9q7eWiAu"  
}
```

SharedFolderMembers

Shared folder user and group membership.

users *List of (UserMembershipInfo)* The list of user members of the shared folder.

groups *List of (GroupMembershipInfo)* The list of group members of the shared folder.

invitees *List of (InviteeMembershipInfo)* The list of invitees to the shared folder.

cursor *String?* Present if there are additional shared folder members that have not been returned yet. Pass the cursor into [list_folder_members/continue](#) to list additional members. This field is optional.

ERRORS

Example: invalid_id

```
{  
    "error_summary": "invalid_id...",  
    "error": {  
        ".tag": "invalid_id"  
    }  
}
```

Example: not_a_member



```
        ".tag": "not_a_member"
    }
```

Example: email_unverified

```
{
    "error_summary": "email_unverified/...",
    "error": {
        ".tag": "email_unverified"
    }
}
```

Example: unmounted

```
{
    "error_summary": "unmounted/...",
    "error": {
        ".tag": "unmounted"
    }
}
```

Example: other

```
{
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

SharedFolderAccessError (*open union*)

There is an error accessing the shared folder. The value will be one of the following datatypes. New values may be introduced as our API evolves.

invalid_id *Void* This shared folder ID is invalid.

not_a_member *Void* The user is not a member of the shared folder thus cannot access it.

email_unverified [deprecated] *Void* Field is deprecated. Never set.

unmounted *Void* The shared folder is unmounted.

/list_folder_members/continue

VERSION

1 ▾

DESCRIPTION Once a cursor has been retrieved from [list_folder_members](#), use this to paginate through all shared folder members.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/list_folder_members/continue

AUTHENTICATION [User Authentication, Dropbox-API-Select-Admin \(Whole Team\)](#)

RPC
FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/list_folder_members/continue \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"cursor\": \"ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu\"}"
```

PARAMETERS

```
{
    "cursor": "ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu"
}
```

ListFolderMembersContinueArg

cursor *String* The cursor returned by your last call to [list_folder_members](#) or [list_folder_members/continue](#).

RETURNS

```
{
    "users": [
        {
            "access_type": {
                ".tag": "owner"
            },
            "user": {
                "account_id": "dbid:AAH4f99T0taONib-OurWxbNQ6ywGRopQngc",
                "email": "bob@example.com",
                "name": {
                    "first": "Bob",
                    "last": "Dylan"
                }
            }
        }
    ]
}
```



≡

```
        "permissions": [],
        "is_inherited": false
    },
    "groups": [
        {
            "access_type": {
                ".tag": "editor"
            },
            "group": {
                "group_name": "Test group",
                "group_id": "g:e2db7665347abcd600000000001a2b3c",
                "group_management_type": {
                    ".tag": "user_managed"
                },
                "group_type": {
                    ".tag": "user_managed"
                },
                "is_member": false,
                "is_owner": false,
                "same_team": true,
                "member_count": 10
            },
            "permissions": [],
            "is_inherited": false
        }
    ],
    "invitees": [
        {
            "access_type": {
                ".tag": "viewer"
            },
            "invitee": {
                ".tag": "email",
                "email": "jessica@example.com"
            },
            "permissions": [],
            "is_inherited": false
        }
    ],
    "cursor": "ZtkX9_EHj3x7PMkVuFIhwKYXEwpwLwyxp9vMKomUhllil9q7eWiAu"
}
```

SharedFolderMembers

Shared folder user and group membership.

users *List of (UserMembershipInfo)* The list of user members of the shared folder.

groups *List of (GroupMembershipInfo)* The list of group members of the shared folder.

invitees *List of (InviteeMembershipInfo)* The list of invitees to the shared folder.

cursor *String?* Present if there are additional shared folder members that have not been returned yet. Pass the cursor into [list_folder_members/continue](#) to list additional members. This field is optional.

ERRORS Example: invalid_cursor

```
{
    "error_summary": "invalid_cursor/...",
    "error": {
        ".tag": "invalid_cursor"
    }
}
```

Example: other

```
{
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

ListFolderMembersContinueError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

access_error *SharedFolderAccessError*

invalid_cursor *Void* ListFolderMembersContinueArg.cursor is invalid.

/list_folders

VERSION

1

DESCRIPTION Return the list of all shared folders the current user has access to.

This endpoint does not support apps with the app folder permission.



AUTHENTICATION

User Authentication

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/list_folders \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"limit\": 100, \"actions\": []}"
```

PARAMETERS

```
{  
    "limit": 100,  
    "actions": []  
}
```

ListFoldersArgs

limit *UInt32(min=1, max=1000)* The maximum number of results to return per request. The default for this field is 1000.

actions *List of (FolderAction)?* A list of `FolderAction`'s corresponding to `FolderPermission`'s that should appear in the response's SharedFolderMetadata.permissions field describing the actions the authenticated user can perform on the folder. This field is optional.

RETURNS

```
{  
    "entries": [  
        {  
            "access_type": {  
                ".tag": "owner"  
            },  
            "is_inside_team_folder": false,  
            "is_team_folder": false,  
            "name": "dir",  
            "policy": {  
                "acl_update_policy": {  
                    ".tag": "owner"  
                },  
                "shared_link_policy": {  
                    ".tag": "anyone"  
                },  
                "member_policy": {  
                    ".tag": "anyone"  
                },  
                "resolved_member_policy": {  
                    ".tag": "team"  
                }  
            },  
            "preview_url": "https://www.dropbox.com/scl/fo/fir9vjelf",  
            "shared_folder_id": "84528192421",  
            "time_invited": "2016-01-20T00:00:00Z",  
            "path_lower": "/dir",  
            "link_metadata": {  
                "audience_options": [  
                    {  
                        ".tag": "public"  
                    },  
                    {  
                        ".tag": "team"  
                    },  
                    {  
                        ".tag": "members"  
                    }  
                ],  
                "current_audience": {  
                    ".tag": "public"  
                },  
                "link_permissions": [  
                    {  
                        "action": {  
                            ".tag": "change_audience"  
                        },  
                        "allow": true  
                    }  
                ],  
                "password_protected": false,  
                "url": ""  
            },  
            "permissions": [],  
            "access_inheritance": {  
                ".tag": "inherit"  
            }  
        },  
        {"cursor": "ZtkX9_EHj3x7PMkVuFIhwKYXEwpwLwyxp9vMKomUhllil9q7eWiAu"}  
    ]  
}
```

ListFoldersResult

Result for [list_folders](#) or [list_mountable_folders](#), depending on which endpoint was requested.

Unmounted shared folders can be identified by the absence of SharedFolderMetadata.path_lower.

entries *List of (SharedFolderMetadata)* List of all shared folders the authenticated user has access to.



[list_mountable_folders/continue](#)) to list additional folders. This field is optional.

ERRORS No errors.

/list_folders/continue

VERSION ▾

DESCRIPTION Once a cursor has been retrieved from [list_folders](#), use this to paginate through all shared folders. The cursor must come from a previous call to [list_folders](#) or [list_folders/continue](#).

This endpoint does not support apps with the app folder permission.

URL STRUCTURE https://api.dropboxapi.com/2/sharing/list_folders/continue

AUTHENTICATION [User Authentication](#)

ENDPOINT FORMAT [RPC](#)

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/list_folders/continue \
  --header "Authorization: Bearer <get access token>" \
  --header "Content-Type: application/json" \
  --data "{\"cursor\": \"ZtkX9_EHj3x7PMkVuFIhwKYXEwpwLwyxp9vMKomUhllil9q7eWiAu\"}"
```

PARAMETERS

```
{\n    \"cursor\": \"ZtkX9_EHj3x7PMkVuFIhwKYXEwpwLwyxp9vMKomUhllil9q7eWiAu\"\n}
```

ListFoldersContinueArg

cursor *String* The cursor returned by the previous API call specified in the endpoint description.

RETURNS

```
{\n    \"entries\": [\n        {\n            \"access_type\": {\n                \".tag\": \"owner\"\n            },\n            \"is_inside_team_folder\": false,\n            \"is_team_folder\": false,\n            \"name\": \"dir\",\n            \"policy\": {\n                \"acl_update_policy\": {\n                    \".tag\": \"owner\"\n                },\n                \"shared_link_policy\": {\n                    \".tag\": \"anyone\"\n                },\n                \"member_policy\": {\n                    \".tag\": \"anyone\"\n                },\n                \"resolved_member_policy\": {\n                    \".tag\": \"team\"\n                }\n            },\n            \"preview_url\": \"https://www.dropbox.com/scl/fo/fir9vjelf\", \n            \"shared_folder_id\": \"84528192421\", \n            \"time_invited\": \"2016-01-20T00:00:00Z\", \n            \"path_lower\": \"/dir\", \n            \"link_metadata\": {\n                \"audience_options\": [\n                    {\n                        \".tag\": \"public\"\n                    },\n                    {\n                        \".tag\": \"team\"\n                    },\n                    {\n                        \".tag\": \"members\"\n                    }\n                ],\n                \"current_audience\": {\n                    \".tag\": \"public\"\n                },\n                \"link_permissions\": [\n                    {\n                        \"action\": {\n                            \".tag\": \"change_audience\"\n                        },\n                        \"allow\": true\n                    }\n                ],\n                \"password_protected\": false,\n                \"url\": \"\"\n            }\n        }\n    ]\n}
```



≡

```
        }
    ],
    "cursor": "ZtkX9_Ehj3x7PMkVuFIhwKYXEwpwLwyxp9vMKomUhllil9q7eWiAu"
}
```

ListFoldersResult

Result for [list_folders](#) or [list_mountable_folders](#), depending on which endpoint was requested.

Unmounted shared folders can be identified by the absence of SharedFolderMetadata.path_lower.

entries *List of (SharedFolderMetadata)* List of all shared folders the authenticated user has access to.

cursor *String?* Present if there are additional shared folders that have not been returned yet. Pass the cursor into the corresponding continue endpoint (either [list_folders/continue](#) or [list_mountable_folders/continue](#)) to list additional folders. This field is optional.

ERRORS Example: invalid_cursor

```
{
  "error_summary": "invalid_cursor/...",
  "error": {
    ".tag": "invalid_cursor"
  }
}
```

Example: other

```
{
  "error_summary": "other/...",
  "error": {
    ".tag": "other"
  }
}
```

ListFoldersContinueError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

invalid_cursor *Void* ListFoldersContinueArg.cursor is invalid.

/list_mountable_folders

VERSION

1 ▾

DESCRIPTION Return the list of all shared folders the current user can mount or unmount.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/list_mountable_folders

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/list_mountable_folders \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"limit\": 100, \"actions\": []}"
```

PARAMETERS

```
{
  "limit": 100,
  "actions": []
}
```

ListFoldersArgs

limit *UInt32(min=1, max=1000)* The maximum number of results to return per request. The default for this field is 1000.

actions *List of (FolderAction)?* A list of `FolderAction`'s corresponding to `FolderPermission`'s that should appear in the response's SharedFolderMetadata.permissions field describing the actions the authenticated user can perform on the folder. This field is optional.

RETURNS

```
{
  "entries": [
```



```
        "is_inside_team_folder": false,
        "is_team_folder": false,
        "name": "dir",
        "policy": {
            "acl_update_policy": {
                ".tag": "owner"
            },
            "shared_link_policy": {
                ".tag": "anyone"
            },
            "member_policy": {
                ".tag": "anyone"
            },
            "resolved_member_policy": {
                ".tag": "team"
            }
        },
        "preview_url": "https://www.dropbox.com/scl/fo/fir9vjelf",
        "shared_folder_id": "84528192421",
        "time_invited": "2016-01-20T00:00:00Z",
        "path_lower": "/dir",
        "link_metadata": {
            "audience_options": [
                {
                    ".tag": "public"
                },
                {
                    ".tag": "team"
                },
                {
                    ".tag": "members"
                }
            ],
            "current_audience": {
                ".tag": "public"
            },
            "link_permissions": [
                {
                    "action": {
                        ".tag": "change_audience"
                    },
                    "allow": true
                }
            ],
            "password_protected": false,
            "url": ""
        },
        "permissions": [],
        "access_inheritance": {
            ".tag": "inherit"
        }
    }
},
"cursor": "ZtkX9_EHj3x7PMkVuFIhwKYXEwpwLwyxp9vMKomUhllil9q7eWiAu"
}
```

ListFoldersResult

Result for [list_folders](#) or [list_mountable_folders](#), depending on which endpoint was requested.

Unmounted shared folders can be identified by the absence of SharedFolderMetadata.path_lower.

entries *List of (SharedFolderMetadata)* List of all shared folders the authenticated user has access to.

cursor *String?* Present if there are additional shared folders that have not been returned yet. Pass the cursor into the corresponding continue endpoint (either [list_folders/continue](#) or [list_mountable_folders/continue](#)) to list additional folders. This field is optional.

ERRORS No errors.

/list_mountable_folders/continue

VERSION

1

DESCRIPTION Once a cursor has been retrieved from [list_mountable_folders](#), use this to paginate through all mountable shared folders. The cursor must come from a previous call to [list_mountable_folders](#) or [list_mountable_folders/continue](#).

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/list_mountable_folders/continue

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE



folders/continue \

```
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"cursor\": \"ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu\"}"
```

PARAMETERS

```
{ "cursor": "ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu" }
```

ListFoldersContinueArg

cursor *String* The cursor returned by the previous API call specified in the endpoint description.

RETURNS

```
{ "entries": [
    {
        "access_type": {
            ".tag": "owner"
        },
        "is_inside_team_folder": false,
        "is_team_folder": false,
        "name": "dir",
        "policy": {
            "acl_update_policy": {
                ".tag": "owner"
            },
            "shared_link_policy": {
                ".tag": "anyone"
            },
            "member_policy": {
                ".tag": "anyone"
            },
            "resolved_member_policy": {
                ".tag": "team"
            }
        },
        "preview_url": "https://www.dropbox.com/scl/fo/fir9vjelf",
        "shared_folder_id": "84528192421",
        "time_invited": "2016-01-20T00:00:00Z",
        "path_lower": "/dir",
        "link_metadata": {
            "audience_options": [
                {
                    ".tag": "public"
                },
                {
                    ".tag": "team"
                },
                {
                    ".tag": "members"
                }
            ],
            "current_audience": {
                ".tag": "public"
            },
            "link_permissions": [
                {
                    "action": {
                        ".tag": "change_audience"
                    },
                    "allow": true
                }
            ],
            "password_protected": false,
            "url": ""
        },
        "permissions": [],
        "access_inheritance": {
            ".tag": "inherit"
        }
    }
],
"cursor": "ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu"
```

ListFoldersResult

Result for [list_folders](#) or [list_mountable_folders](#), depending on which endpoint was requested.

Unmounted shared folders can be identified by the absence of SharedFolderMetadata.path_lower.

entries *List of (SharedFolderMetadata)* List of all shared folders the authenticated user has access to.

cursor *String?* Present if there are additional shared folders that have not been returned yet. Pass the cursor into the corresponding continue endpoint (either [list_folders/continue](#) or [list_mountable_folders/continue](#)) to list additional folders. This field is optional.

ERRORS

Example: invalid_cursor

```
{
    "error_summary": "invalid_cursor/...",
    "error": {
        ".tag": "invalid_cursor"
    }
}
```



Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

ListFoldersContinueError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

invalid_cursor *Void* ListFoldersContinueArg.cursor is invalid.

/list_received_files

VERSION

1 ▾

DESCRIPTION Returns a list of all files shared with current user.
Does not include files the user has received via shared folders, and does not include unclaimed invitations.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/list_received_files

AUTHENTICATION [User Authentication, Dropbox-API-Select-Admin \(Team Admin\)](#)

ENDPOINT FORMAT
[RPC](#)

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/list_received_files \  
--header "Authorization: Bearer <get access token>" \  
--header "Content-Type: application/json" \  
--data "{\"limit\": 100,\"actions\": []}"
```

PARAMETERS

```
{  
    "limit": 100,  
    "actions": []  
}
```

ListFilesArg

Arguments for [list_received_files](#).

limit *UInt32(min=1, max=300)* Number of files to return max per query. Defaults to 100 if no limit is specified. The default for this field is 100.

actions *List of (FileAction)?* A list of `FileAction`'s corresponding to `FilePermission`'s that should appear in the response's SharedFileMetadata.permissions field describing the actions the authenticated user can perform on the file. This field is optional.

RETURNS

```
{  
    "entries": [  
        {  
            "id": "id:3kmLmQFnf1AAAAAAAAAAw",  
            "name": "file.txt",  
            "policy": {  
                "acl_update_policy": {  
                    ".tag": "owner"  
                },  
                "shared_link_policy": {  
                    ".tag": "anyone"  
                },  
                "member_policy": {  
                    ".tag": "anyone"  
                },  
                "resolved_member_policy": {  
                    ".tag": "team"  
                }  
            },  
            "preview_url": "https://www.dropbox.com/scl/fi/fir9vjelf",  
            "access_type": {  
                ".tag": "viewer"  
            },  
            "owner_display_names": [  
                "Jane Doe"  
            ],  
            "owner_team": {  
                "id": "dbtid:AAFdgehTzw7WlXhZJsbGCLePe8RvQGYDr-I",  
                "name": "Acme, Inc."  
            },  
            "path_display": "/dir/file.txt",  
            "path_lower": "/dir/file.txt",  
            "size": 1234567890  
        }  
    ]  
}
```



≡

```
    "cursor": "AzJJbGlzdF90eXBdofe9c3RPbGlzdGFyZ3NfYnlfZ2lkMRhcbric7Rdog9cmV2aXNpb24H3Qf6o1fkHxQ"
```

ListFilesResult

Success results for [list_received_files](#).

entries *List of (SharedFileMetadata)* Information about the files shared with current user.

cursor *String?* Cursor used to obtain additional shared files. This field is optional.

ERRORS Example: [email_unverified](#)

```
{  
  "error_summary": "email_unverified/...",  
  "error": {  
    ".tag": "email_unverified"  
  }  
}
```

Example: [other](#)

```
{  
  "error_summary": "other/...",  
  "error": {  
    ".tag": "other"  
  }  
}
```

SharingUserError (*open union*)

User account had a problem preventing this action. The value will be one of the following datatypes. New values may be introduced as our API evolves.

email_unverified *Void* The current user must verify the account email address before performing this action.

/list_received_files/continue

VERSION

1 ▾

DESCRIPTION Get more results with a cursor from [list_received_files](#).

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

```
https://api.dropboxapi.com/2/sharing/list_received_files/continue
```

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/list_received_files/continue \  
  --header "Authorization: Bearer <get access token>" \  
  --header "Content-Type: application/json" \  
  --data "{\"cursor\": \"AzJJbGlzdF90eXBdofe9c3RPbGlzdGFyZ3NfYnlfZ2lkMRhcbric7Rdog9emfGRlc2MCRWxpbwL0BGRId\"}"
```

PARAMETERS

```
{  
  "cursor": "AzJJbGlzdF90eXBdofe9c3RPbGlzdGFyZ3NfYnlfZ2lkMRhcbric7Rdog9emfGRlc2MCRWxpbwL0BGRId"  
}
```

ListFilesContinueArg

Arguments for [list_received_files/continue](#).

cursor *String* Cursor in ListFilesResult.cursor.

RETURNS

```
{  
  "entries": [  
    {  
      "id": "id:3kmLmQFnf1AAAAAAAAAAw",  
      "name": "file.txt",  
      "policy": {  
        "acl_update_policy": {  
          ".tag": "owner"  
        },  
        "shared_link_policy": {  
          ".tag": "anyone"  
        },  
        "member_policy": {  
          ".tag": "anyone"  
        },  
        "resolved_member_policy": {  
          ".tag": "team"  
        }  
      }  
    }  
  ]  
}
```



≡ self,

```
        ".tag": "viewer"
    },
    "owner_display_names": [
        "Jane Doe"
    ],
    "owner_team": {
        "id": "dbtid:AAFdgehTzwWlXhZJsbGCLePe8RvQGYDr-I",
        "name": "Acme, Inc."
    },
    "path_display": "/dir/file.txt",
    "path_lower": "/dir/file.txt",
    "permissions": [],
    "time_invited": "2016-01-20T00:00:00Z"
],
"cursor": "AzJJbGlzdF90eXBdofe9c3RPbGlzdGFyZ3NfYnlfZ2lkMRhcbric7Rdog9cmV2aNpb24H3Qf6o1fkHxQ"
```

ListFilesResult

Success results for [list_received_files](#).

entries *List of (SharedFileMetadata)* Information about the files shared with current user.

cursor *String?* Cursor used to obtain additional shared files. This field is optional.

ERRORS Example: invalid_cursor

```
{
    "error_summary": "invalid_cursor/...",
    "error": {
        ".tag": "invalid_cursor"
    }
}
```

Example: other

```
{
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

ListFilesContinueError (*open union*)

Error results for [list_received_files/continue](#). The value will be one of the following datatypes. New values may be introduced as our API evolves.

user_error *SharingUserError* User account had a problem.

invalid_cursor *Void* ListFilesContinueArg.cursor is invalid.

/list_shared_links

VERSION

1 ▾

DESCRIPTION

List shared links of this user.

If no path is given, returns a list of all shared links for the current user.

If a non-empty path is given, returns a list of all shared links that allow access to the given path - direct links to the given path and links to parent folders of the given path. Links to parent folders can be suppressed by setting direct_only to true.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/list_shared_links

AUTHENTICATION

[User Authentication](#)

RPC
FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/list_shared_links \
    -header "Authorization: Bearer <get access token>" \
    -header "Content-Type: application/json" \
    --data "{\"cursor\": \"ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu\"}"
```

PARAMETERS

Example: List all links.

```
{
    "cursor": "ZtkX9_EHj3x7PMkVuFIhwKYXEpwdLwyxp9vMKomUhllil9q7eWiAu"
```

Example: path



≡

Example: id

```
{  
    "path": "id:a4ayc_80_0EAAAAAAAAYa"  
}
```

Example: rev

```
{  
    "path": "rev:alc10ce0dd78"  
}
```

Example: id_no_parent_links

```
{  
    "path": "id:a4ayc_80_0EAAAAAAAAYa",  
    "direct_only": true  
}
```

ListSharedLinksArg

path *String(pattern="(.|[r\nn])*|id:.*|(rev:[0-9a-f]{9,})|(ns:[0-9]+(.*)?)")?* See [list_shared_links](#) description. This field is optional.

cursor *String?* The cursor returned by your last call to [list_shared_links](#). This field is optional.

direct_only *Boolean?* See [list_shared_links](#) description. This field is optional.

RETURNS

```
{  
    "links": [  
        {  
            ".tag": "file",  
            "url": "https://www.dropbox.com/s/2sn712vylovegw8/Prime_Numbers.txt?dl=0",  
            "name": "Prime_Numbers.txt",  
            "link_permissions": {  
                "can_revoke": false,  
                "resolved_visibility": {  
                    ".tag": "public"  
                },  
                "revoke_failure_reason": {  
                    ".tag": "owner_only"  
                }  
            },  
            "client_modified": "2015-05-12T15:50:38Z",  
            "server_modified": "2015-05-12T15:50:38Z",  
            "rev": "alc10ce0dd78",  
            "size": 7212,  
            "id": "id:a4ayc_80_0EAAAAAAAAXw",  
            "path_lower": "/homework/math/prime_numbers.txt",  
            "team_member_info": {  
                "team_info": {  
                    "id": "dbtid:AAFdgehTzw7WlXhZJsbGCLePe8RvQGYDr-I",  
                    "name": "Acme, Inc."  
                },  
                "display_name": "Roger Rabbit",  
                "member_id": "dbmid:abcd1234"  
            }  
        },  
        {  
            "has_more": true,  
            "cursor": "ZtkX9_EHj3x7PMkVuFIhwKYXEwpwLwyxp9vMKomUhllil9q7eWiAu"  
        }  
    ]  
}
```

ListSharedLinksResult

links *List of (SharedLinkMetadata)* Shared links applicable to the path argument.

has_more *Boolean* Is true if there are additional shared links that have not been returned yet. Pass the cursor into [list_shared_links](#) to retrieve them.

cursor *String?* Pass the cursor into [list_shared_links](#) to obtain the additional links. Cursor is returned only if no path is given. This field is optional.

ERRORS

Example: reset

```
{  
    "error_summary": "reset/...",  
    "error": {  
        ".tag": "reset"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```



LISTSHAREDLINKSError (open union)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path *LookupError*

reset *Void* Indicates that the cursor has been invalidated. Call [list_shared_links](#) to obtain a new cursor.

/modify_shared_link_settings

VERSION

1

DESCRIPTION

Modify the shared link's settings.

If the requested visibility conflict with the shared links policy of the team or the shared folder (in case the linked file is part of a shared folder) then the LinkPermissions.resolved_visibility of the returned SharedLinkMetadata will reflect the actual visibility of the shared link and the LinkPermissions.requested_visibility will reflect the requested visibility.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/modify_shared_link_settings

AUTHENTICATION

[User Authentication](#)

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/sharing/modify_shared_link_settings \
  --header "Authorization: Bearer <get access token>" \
  --header "Content-Type: application/json" \
  --data "{\"url\": \"https://www.dropbox.com/s/2sn712vylovezg8/Prime_Numbers.txt?dl=0\", \"settings\": {\"requested_visibility\": \"public\", \"audience\": \"public\", \"access\": \"editor\"}, \"remove_expiration\": false}
```

PARAMETERS

```
{ "url": "https://www.dropbox.com/s/2sn712vylovezg8/Prime_Numbers.txt?dl=0",
  "settings": {
    "requested_visibility": "public",
    "audience": "public",
    "access": "editor"
  },
  "remove_expiration": false
}
```

ModifySharedLinkSettingsArgs

url *String* URL of the shared link to change its settings.

settings *SharedLinkSettings* Set of settings for the shared link.

remove_expiration *Boolean* If set to true, removes the expiration of the shared link. The default for this field is False.

RETURNS

```
{ ".tag": "file",
  "url": "https://www.dropbox.com/s/2sn712vylovezg8/Prime_Numbers.txt?dl=0",
  "name": "Prime_Numbers.txt",
  "link_permissions": {
    "can_revoke": false,
    "resolved_visibility": {
      ".tag": "public"
    },
    "revoke_failure_reason": {
      ".tag": "owner_only"
    }
  },
  "client_modified": "2015-05-12T15:50:38Z",
  "server_modified": "2015-05-12T15:50:38Z",
  "rev": "alcl0ce0dd78",
  "size": 7212,
  "id": "id:a4ayc_80_0EAAAAAAAAXw",
  "path_lower": "/homework/math/prime_numbers.txt",
  "team_member_info": {
    "team_info": {
      "id": "dbtid:AAFdgehTzw7WlXhZJsbGCLePe8RvQGYDr-I",
      "name": "Acme, Inc."
    },
    "display_name": "Roger Rabbit",
    "member_id": "dbmid:abcd1234"
  }
}
```

```
{ ".tag": "folder",
  "url": "https://www.dropbox.com/sh/s6fvw6ol7rmqolx/AAAgWRShjmYDvPpDB30Sykjfa?dl=0",
  "name": "Math",
  "link_permissions": {
    "can_revoke": false,
    "resolved_visibility": {
      ".tag": "public"
    },
    "revoke_failure_reason": {
      ".tag": "owner_only"
    }
  }
}
```



```
"path_lower": "/homework/math",
"team_member_info": {
    "team_info": {
        "id": "dbtid:AAFdgehTzw7WlXhZJsbGCLePe8RvQGYDr-I",
        "name": "Acme, Inc."
    },
    "display_name": "Roger Rabbit",
    "member_id": "dbmid:abcd1234"
}
```

SharedLinkMetadata (*datatype with subtypes*)

The metadata of a shared link. This datatype will be one of the following subtypes:

file *FileLinkMetadata*

folder *FolderLinkMetadata*

ERRORS Example: `shared_link_not_found`

```
{
    "error_summary": "shared_link_not_found/...",
    "error": {
        ".tag": "shared_link_not_found"
    }
}
```

Example: `shared_link_access_denied`

```
{
    "error_summary": "shared_link_access_denied/...",
    "error": {
        ".tag": "shared_link_access_denied"
    }
}
```

Example: `unsupported_link_type`

```
{
    "error_summary": "unsupported_link_type/...",
    "error": {
        ".tag": "unsupported_link_type"
    }
}
```

Example: `other`

```
{
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

Example: `email_not_verified`

```
{
    "error_summary": "email_not_verified/...",
    "error": {
        ".tag": "email_not_verified"
    }
}
```

ModifySharedLinkSettingsError (*union*)

The value will be one of the following datatypes:

shared_link_not_found *Void* The shared link wasn't found.

shared_link_access_denied *Void* The caller is not allowed to access this shared link.

unsupported_link_type *Void* This type of link is not supported; use [files/export](#) instead.

settings_error *SharedLinkSettingsError* There is an error with the given settings.

email_not_verified *Void* The caller's email should be verified.

/mount_folder

VERSION

1

DESCRIPTION

The current user mounts the designated folder.

Mount a shared folder for a user after they have been added as a member. Once mounted, the shared folder will appear in their Dropbox.

This endpoint does not support apps with the app folder permission.



AUTHENTICATION

User Authentication

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/sharing/mount_folder \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"shared_folder_id\": \"84528192421\"}"
```

PARAMETERS

```
{  
  "shared_folder_id": "84528192421"  
}
```

MountFolderArg

shared_folder_id *String(pattern="[-_0-9a-zA-Z:]")* The ID of the shared folder to mount.

RETURNS

```
{  
  "access_type": {  
    ".tag": "owner"  
  },  
  "is_inside_team_folder": false,  
  "is_team_folder": false,  
  "name": "dir",  
  "policy": {  
    "acl_update_policy": {  
      ".tag": "owner"  
    },  
    "shared_link_policy": {  
      ".tag": "anyone"  
    },  
    "member_policy": {  
      ".tag": "anyone"  
    },  
    "resolved_member_policy": {  
      ".tag": "team"  
    }  
  },  
  "preview_url": "https://www.dropbox.com/scl/fo/fir9vjelf",  
  "shared_folder_id": "84528192421",  
  "time_invited": "2016-01-20T00:00:00Z",  
  "path_lower": "/dir",  
  "link_metadata": {  
    "audience_options": [  
      {  
        ".tag": "public"  
      },  
      {  
        ".tag": "team"  
      },  
      {  
        ".tag": "members"  
      }  
    ],  
    "current_audience": {  
      ".tag": "public"  
    },  
    "link_permissions": [  
      {  
        "action": {  
          ".tag": "change_audience"  
        },  
        "allow": true  
      }  
    ],  
    "password_protected": false,  
    "url": ""  
  },  
  "permissions": [],  
  "access_inheritance": {  
    ".tag": "inherit"  
  }  
}
```

SharedFolderMetadata

The metadata which includes basic information about the shared folder.

access_type *AccessLevel* The current user's access level for this shared folder.

is_inside_team_folder *Boolean* Whether this folder is inside of a team folder.

is_team_folder *Boolean* Whether this folder is a [team folder](#).

name *String* The name of the this shared folder.

policy *FolderPolicy* Policies governing this shared folder.

preview_url *String* URL for displaying a web preview of the shared folder.

shared_folder_id *String(pattern="[-_0-9a-zA-Z:]")* The ID of the shared folder.



owner_display_names *List of (String)?* The display names of the users that own the folder. If the folder is part of a team folder, the display names of the team admins are also included. Absent if the owner display names cannot be fetched. This field is optional.

owner_team *Team?* The team that owns the folder. This field is not present if the folder is not owned by a team. This field is optional.

parent_shared_folder_id *String(pattern="[-_0-9a-zA-Z]+")?* The ID of the parent shared folder. This field is present only if the folder is contained within another shared folder. This field is optional.

path_lower *String?* The lower-cased full path of this shared folder. Absent for unmounted folders. This field is optional.

parent_folder_name *String?* Display name for the parent folder. This field is optional.

link_metadata *SharedContentLinkMetadata?* The metadata of the shared content link to this shared folder. Absent if there is no link on the folder. This is for an unreleased feature so it may not be returned yet. This field is optional.

permissions *List of (FolderPermission)?* Actions the current user may perform on the folder and its contents. The set of permissions corresponds to the FolderActions in the request. This field is optional.

access_inheritance *AccessInheritance* Whether the folder inherits its members from its parent. The default for this union is inherit.

ERRORS Example: inside_shared_folder

```
{  
    "error_summary": "inside_shared_folder/...",  
    "error": {  
        ".tag": "inside_shared_folder"  
    }  
}
```

Example: already_mounted

```
{  
    "error_summary": "already_mounted/...",  
    "error": {  
        ".tag": "already_mounted"  
    }  
}
```

Example: no_permission

```
{  
    "error_summary": "no_permission/...",  
    "error": {  
        ".tag": "no_permission"  
    }  
}
```

Example: not_mountable

```
{  
    "error_summary": "not_mountable/...",  
    "error": {  
        ".tag": "not_mountable"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

MountFolderError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

access_error *SharedFolderAccessError*

inside_shared_folder *Void* Mounting would cause a shared folder to be inside another, which is disallowed.

insufficient_quota *InsufficientQuotaAmounts* The current user does not have enough space to mount the shared folder.

already_mounted *Void* The shared folder is already mounted.

no_permission *Void* The current user does not have permission to perform this action.

not_mountable *Void* The shared folder is not mountable. One example where this can occur is when the shared folder belongs



VERSION

1

DESCRIPTION	The current user relinquishes their membership in the designated file. Note that the current user may still have inherited access to this file through the parent folder.
-------------	---

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/relinquish_file_membership

AUTHENTICATION

[User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/sharing/relinquish_file_membership \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"file\": \"id:3kmLmQFnf1AAAAAAAAsAw\"}"
```

PARAMETERS

```
{  
    "file": "id:3kmLmQFnf1AAAAAAAAsAw"  
}
```

RelinquishFileMembershipArg

file String(min_length=1, pattern="(/{id:}.*|nspath:[0-9]+:{})|ns:[0-9]+({\.*})?") The path or id for the file.

RETURNS

No return values.

ERRORS

Example: group_access

```
{  
    "error_summary": "group_access/...",  
    "error": {  
        ".tag": "group_access"  
    }  
}
```

Example: no_permission

```
{  
    "error_summary": "no_permission/...",  
    "error": {  
        ".tag": "no_permission"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

RelinquishFileMembershipError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

access_error *Sharing FileAccess Error*

group_access *Void* The current user has access to the shared file via a group. You can't relinquish membership to a file shared via groups.

no_permission *Void* The current user does not have permission to perform this action.

/relinquish_folder_membership

VERSION

1

DESCRIPTION

The current user relinquishes their membership in the designated shared folder and will no longer have access to the folder. A folder owner cannot relinquish membership in their own folder.



is



This endpoint does not support apps with the app folder permission.

URL STRUCTURE

```
https://api.dropboxapi.com/2/sharing/relinquish_folder_membership
```

AUTHENTICATION

User Authentication

ENDPOINT FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/relinquish_folder_membership \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"shared_folder_id\": \"84528192421\", \"leave_a_copy\": false}"
```

PARAMETERS

```
{  
    "shared_folder_id": "84528192421",  
    "leave_a_copy": false  
}
```

RelinquishFolderMembershipArg

shared_folder_id *String(pattern="[-_0-9a-zA-Z:]")* The ID for the shared folder.

leave_a_copy *Boolean* Keep a copy of the folder's contents upon relinquishing membership. The default for this field is False.

RETURNS

Example: complete

```
{  
    ".tag": "complete"  
}
```

Example: async_job_id

```
{  
    ".tag": "async_job_id",  
    "async_job_id": "34g93hh34h04y384084"  
}
```

LaunchEmptyResult (*union*)

Result returned by methods that may either launch an asynchronous job or complete synchronously. Upon synchronous completion of the job, no additional information is returned. This datatype comes from an imported namespace originally defined in the `async` namespace. The value will be one of the following datatypes:

async_job_id *String(min_length=1)* This response indicates that the processing is asynchronous. The string is an id that can be used to obtain the status of the asynchronous job.

complete *Void* The job finished synchronously and successfully.

ERRORS

Example: folder_owner

```
{  
    "error_summary": "folder_owner/...",  
    "error": {  
        ".tag": "folder_owner"  
    }  
}
```

Example: mounted

```
{  
    "error_summary": "mounted/...",  
    "error": {  
        ".tag": "mounted"  
    }  
}
```

Example: group_access

```
{  
    "error_summary": "group_access/...",  
    "error": {  
        ".tag": "group_access"  
    }  
}
```

Example: team_folder

```
{  
    "error_summary": "team_folder/...",  
}
```



≡

Example: no_permission

```
{  
    "error_summary": "no_permission/...",  
    "error": {  
        ".tag": "no_permission"  
    }  
}
```

Example: no_explicit_access

```
{  
    "error_summary": "no_explicit_access/...",  
    "error": {  
        ".tag": "no_explicit_access"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

RelinquishFolderMembershipError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

access_error *SharedFolderAccessError*

folder_owner *Void* The current user is the owner of the shared folder. Owners cannot relinquish membership to their own folders. Try unsharing or transferring ownership first.

mounted *Void* The shared folder is currently mounted. Unmount the shared folder before relinquishing membership.

group_access *Void* The current user has access to the shared folder via a group. You can't relinquish membership to folders shared via groups.

team_folder *Void* This action cannot be performed on a team shared folder.

no_permission *Void* The current user does not have permission to perform this action.

no_explicit_access *Void* The current user only has inherited access to the shared folder. You can't relinquish inherited membership to folders.

/remove_file_member_2

VERSION

1

DESCRIPTION

Removes a specified member from the file.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/remove_file_member_2

AUTHENTICATION

[User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/remove_file_member_2 \  
    -header "Authorization: Bearer <get access token>" \  
    -header "Content-Type: application/json" \  
    -data "{\"file\": \"id:3kmLmQFnf1AAAAAAAAAAw\", \"member\": {\".tag\": \"email\", \"email\": \"justin@example.com\"}}
```

PARAMETERS

```
{  
    "file": "id:3kmLmQFnf1AAAAAAAAAAw",  
    "member": {  
        ".tag": "email",  
        "email": "justin@example.com"  
    }  
}
```

[RemoveFileMemberArg](#)

Arguments for [remove_file_member_2](#).



is:

that

even if an email is specified, it may result in the removal of a user (not an invitee) if the user's main account corresponds to that email address.

RETURNS FileMemberRemoveActionResult (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

success MemberAccessLevelResult Member was successfully removed from this file.

member_error FileMemberActionError User was not able to remove this member.

ERRORS RemoveFileMemberError (*open union*)

Errors for [remove_file_member_2](#). The value will be one of the following datatypes. New values may be introduced as our API evolves.

user_error SharingUserError

access_error Sharing FileAccess Error

no_explicit_access MemberAccessLevelResult This member does not have explicit access to the file and therefore cannot be removed. The return value is the access that a user might have to the file from a parent folder.

/remove_folder_member

VERSION

1 ▾

DESCRIPTION Allows an owner or editor (if the ACL update policy allows) of a shared folder to remove another member.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/remove_folder_member

AUTHENTICATION

[User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/remove_folder_member \
  --header "Authorization: Bearer <get access token>" \
  --header "Content-Type: application/json" \
  --data "{\"shared_folder_id\": \"84528192421\", \"member\": {\"tag\": \"email\", \"email\": \"justin@example.com\"}, \"leave_a_copy\": false}
```

PARAMETERS

```
{\n    \"shared_folder_id\": \"84528192421\", \n    \"member\": {\n        \"tag\": \"email\", \n        \"email\": \"justin@example.com\"\n    }, \n    \"leave_a_copy\": false\n}
```

RemoveFolderMemberArg

shared_folder_id String(*pattern*=“[-_0-9a-zA-Z:]”+) The ID for the shared folder.

member MemberSelector The member to remove from the folder.

leave_a_copy Boolean If true, the removed user will keep their copy of the folder after it's unshared, assuming it was mounted. Otherwise, it will be removed from their Dropbox. Also, this must be set to false when kicking a group.

RETURNS

```
{\n    \"tag\": \"async_job_id\", \n    \"async_job_id\": \"34g93hh34h04y384084\"\n}
```

LaunchResultBase (*union*)

Result returned by methods that launch an asynchronous job.

A method who may either launch an asynchronous job, or complete the request synchronously, can use this union by extending it, and adding a 'complete' field with the type of the synchronous response. See [LaunchEmptyResult](#) for an example. This datatype comes from an imported namespace originally defined in the `async` namespace. The value will be one of the following datatypes:



ERRORS Example: folder_owner

```
{  
    "error_summary": "folder_owner/...",  
    "error": {  
        ".tag": "folder_owner"  
    }  
}
```

Example: group_access

```
{  
    "error_summary": "group_access/...",  
    "error": {  
        ".tag": "group_access"  
    }  
}
```

Example: team_folder

```
{  
    "error_summary": "team_folder/...",  
    "error": {  
        ".tag": "team_folder"  
    }  
}
```

Example: no_permission

```
{  
    "error_summary": "no_permission/...",  
    "error": {  
        ".tag": "no_permission"  
    }  
}
```

Example: too_many_files

```
{  
    "error_summary": "too_many_files/...",  
    "error": {  
        ".tag": "too_many_files"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

RemoveFolderMemberError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

access_error *SharedFolderAccessError*

member_error *SharedFolderMemberError*

folder_owner *Void* The target user is the owner of the shared folder. You can't remove this user until ownership has been transferred to another member.

group_access *Void* The target user has access to the shared folder via a group.

team_folder *Void* This action cannot be performed on a team shared folder.

no_permission *Void* The current user does not have permission to perform this action.

too_many_files *Void* This shared folder has too many files for leaving a copy. You can still remove this user without leaving a copy.

/revoke_shared_link

VERSION

1

DESCRIPTION

Revoke a shared link.

Note that even after revoking a shared link to a file, the file may be accessible if there are shared links leading to any of the file parent folders. To list all shared links that enable access to a specific file, you can use the [list_shared_links](#) with the file as the ListSharedLinksArg.path argument.



AUTHENTICATION

User Authentication, Dropbox-API-Select-Admin (Team Admin)

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/revoke_shared_link \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"url\": \"https://www.dropbox.com/s/2sn712vy1ovegw8/Prime_Numbers.txt?dl=0\"}"
```

PARAMETERS

```
{  
  "url": "https://www.dropbox.com/s/2sn712vy1ovegw8/Prime_Numbers.txt?dl=0"  
}
```

RevokeSharedLinkArg

url *String* URL of the shared link.

RETURNS

No return values.

ERRORS

Example: shared_link_not_found

```
{  
  "error_summary": "shared_link_not_found/...",  
  "error": {  
    ".tag": "shared_link_not_found"  
  }  
}
```

Example: shared_link_access_denied

```
{  
  "error_summary": "shared_link_access_denied/...",  
  "error": {  
    ".tag": "shared_link_access_denied"  
  }  
}
```

Example: unsupported_link_type

```
{  
  "error_summary": "unsupported_link_type/...",  
  "error": {  
    ".tag": "unsupported_link_type"  
  }  
}
```

Example: other

```
{  
  "error_summary": "other/...",  
  "error": {  
    ".tag": "other"  
  }  
}
```

Example: shared_link_malformed

```
{  
  "error_summary": "shared_link_malformed/...",  
  "error": {  
    ".tag": "shared_link_malformed"  
  }  
}
```

RevokeSharedLinkError (*union*)

The value will be one of the following datatypes:

shared_link_not_found *Void* The shared link wasn't found.**shared_link_access_denied** *Void* The caller is not allowed to access this shared link.**unsupported_link_type** *Void* This type of link is not supported; use [files/export](#) instead.**shared_link_malformed** *Void* Shared link is malformed.

/set_access_inheritance

VERSION

1 ▾

DESCRIPTION

Change the inheritance policy of an existing Shared Folder. Only permitted for shared folders in a shared team root.



This endpoint does not support apps with the app folder permission.

URL STRUCTURE

```
https://api.dropboxapi.com/2/sharing/set_access_inheritance
```

AUTHENTICATION

User Authentication

ENDPOINT FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/set_access_inheritance \
  --header "Authorization: Bearer <get access token>" \
  --header "Content-Type: application/json" \
  --data "{\"shared_folder_id\": \"84528192421\", \"access_inheritance\": \"inherit\"}"
```

PARAMETERS

```
{  
  "shared_folder_id": "84528192421",  
  "access_inheritance": "inherit"  
}
```

SetAccessInheritanceArg

shared_folder_id *String(pattern="[-_0-9a-zA-Z:]")* The ID for the shared folder.

access_inheritance *AccessInheritance* The access inheritance settings for the folder. The default for this union is inherit.

RETURNS

```
{  
  ".tag": "complete",  
  "access_type": {  
    ".tag": "owner"  
  },  
  "is_inside_team_folder": false,  
  "is_team_folder": false,  
  "name": "dir",  
  "policy": {  
    "acl_update_policy": {  
      ".tag": "owner"  
    },  
    "shared_link_policy": {  
      ".tag": "anyone"  
    },  
    "member_policy": {  
      ".tag": "anyone"  
    },  
    "resolved_member_policy": {  
      ".tag": "team"  
    }  
  },  
  "preview_url": "https://www.dropbox.com/scl/fo/fir9vjelf",  
  "shared_folder_id": "84528192421",  
  "time_invited": "2016-01-20T00:00:00Z",  
  "path_lower": "/dir",  
  "link_metadata": {  
    "audience_options": [  
      {  
        ".tag": "public"  
      },  
      {  
        ".tag": "team"  
      },  
      {  
        ".tag": "members"  
      }  
    ],  
    "current_audience": {  
      ".tag": "public"  
    },  
    "link_permissions": [  
      {  
        "action": {  
          ".tag": "change_audience"  
        },  
        "allow": true  
      }  
    ],  
    "password_protected": false,  
    "url": ""  
  },  
  "permissions": [],  
  "access_inheritance": {  
    ".tag": "inherit"  
  }  
}
```

ShareFolderLaunch (union)

The value will be one of the following datatypes:

async_job_id *String(min_length=1)* This response indicates that the processing is asynchronous. The string is an id that can be used to obtain the status of the asynchronous job.



```
{
  "error_summary": "no_permission/...",
  "error": {
    ".tag": "no_permission"
  }
}
```

Example: no_permission

```
{
  "error_summary": "no_permission/...",
  "error": {
    ".tag": "no_permission"
  }
}
```

Example: other

```
{
  "error_summary": "other/...",
  "error": {
    ".tag": "other"
  }
}
```

SetAccessInheritanceError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

access_error *SharedFolderAccessError* Unable to access shared folder.

no_permission *Void* The current user does not have permission to perform this action.

/share_folder

VERSION

1

DESCRIPTION

Share a folder with collaborators.

Most sharing will be completed synchronously. Large folders will be completed

asynchronously. To make testing the async case repeatable, set

`ShareFolderArg.force_async`.

If a ShareFolderLaunch.async_job_id is returned, you'll need to call [check_share_job_status](#) until the action completes to get the metadata for the folder.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/share_folder

AUTHENTICATION

[User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/sharing/share_folder \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"path\": \"/example/workspace\", \"acl_update_policy\": \"editors\", \"force_async\": false, \"member_pc\"}
```

PARAMETERS

```
{
  "path": "/example/workspace",
  "acl_update_policy": "editors",
  "force_async": false,
  "member_policy": "team",
  "shared_link_policy": "members",
  "access_inheritance": "inherit"
```

ShareFolderArg

path *String(pattern="(.|[r|n])*|(ns:[0-9]+(.*)?)"')* The path to the folder to share. If it does not exist, then a new one is created.

acl_update_policy *AclUpdatePolicy?* Who can add and remove members of this shared folder. This field is optional.

force_async *Boolean* Whether to force the share to happen asynchronously. The default for this field is False.

member_policy *MemberPolicy?* Who can be a member of this shared folder. Only applicable if the current user is on a team. This field is optional.



SharedLinkPolicy.members. This field is optional.

viewer_info_policy *ViewerInfoPolicy?* Who can enable/disable viewer info for this shared folder. This field is optional.

access_inheritance *AccessInheritance* The access inheritance settings for the folder. The default for this union is inherit.

actions *List of (FolderAction)?* A list of `FolderAction`'s corresponding to `FolderPermission`'s that should appear in the response's SharedFolderMetadata.permissions field describing the actions the authenticated user can perform on the folder. This field is optional.

link_settings *LinkSettings?* Settings on the link for this folder. This field is optional.

RETURNS

```
{  
    ".tag": "complete",  
    "access_type": {  
        ".tag": "owner"  
    },  
    "is_inside_team_folder": false,  
    "is_team_folder": false,  
    "name": "dir",  
    "policy": {  
        "acl_update_policy": {  
            ".tag": "owner"  
        },  
        "shared_link_policy": {  
            ".tag": "anyone"  
        },  
        "member_policy": {  
            ".tag": "anyone"  
        },  
        "resolved_member_policy": {  
            ".tag": "team"  
        }  
    },  
    "preview_url": "https://www.dropbox.com/scl/fo/fir9vjelf",  
    "shared_folder_id": "84528192421",  
    "time_invited": "2016-01-20T00:00:00Z",  
    "path_lower": "/dir",  
    "link_metadata": {  
        "audience_options": [  
            {  
                ".tag": "public"  
            },  
            {  
                ".tag": "team"  
            },  
            {  
                ".tag": "members"  
            }  
        ],  
        "current_audience": {  
            ".tag": "public"  
        },  
        "link_permissions": [  
            {  
                "action": {  
                    ".tag": "change_audience"  
                },  
                "allow": true  
            }  
        ],  
        "password_protected": false,  
        "url": ""  
    },  
    "permissions": [],  
    "access_inheritance": {  
        ".tag": "inherit"  
    }  
}
```

ShareFolderLaunch (*union*)

The value will be one of the following datatypes:

async_job_id *String(min_length=1)* This response indicates that the processing is asynchronous. The string is an id that can be used to obtain the status of the asynchronous job.

complete *SharedFolderMetadata*

ERRORS

Example: email_unverified

```
{  
    "error_summary": "email_unverified/...",  
    "error": {  
        ".tag": "email_unverified"  
    }  
}
```

Example: team_policy_disallows_member_policy

```
{  
    "error_summary": "team_policy_disallows_member_policy/...",  
    "error": {  
        ".tag": "team_policy_disallows_member_policy"  
    }  
}
```



≡



Example: disallowed_shared_link_policy

```
{  
    "error_summary": "disallowed_shared_link_policy/...",  
    "error": {  
        ".tag": "disallowed_shared_link_policy"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: no_permission

```
{  
    "error_summary": "no_permission/...",  
    "error": {  
        ".tag": "no_permission"  
    }  
}
```

ShareFolderError (*union*)

The value will be one of the following datatypes:

email_unverified *Void* The current user's e-mail address is unverified.

bad_path *SharePathError* ShareFolderArg.path is invalid.

team_policy_disallows_member_policy *Void* Team policy is more restrictive than ShareFolderArg.member_policy.

disallowed_shared_link_policy *Void* The current user's account is not allowed to select the specified ShareFolderArg.shared_link_policy.

no_permission *Void* The current user does not have permission to perform this action.

/transfer_folder

VERSION

1

DESCRIPTION

Transfer ownership of a shared folder to a member of the shared folder.

User must have AccessLevel.owner access to the shared folder to perform a transfer.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/transfer_folder

AUTHENTICATION

[User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot



```
curl -X POST https://api.dropboxapi.com/2/sharing/transfer_folder \  
    -header "Authorization: Bearer <get access token>" \  
    -header "Content-Type: application/json" \  
    -data "{\"shared_folder_id\": \"84528192421\", \"to_dropbox_id\": \"dbid:AAEufNrMPSPe0dMQijRP0N_aZtBJRm26W4Q\"}"
```

PARAMETERS

```
{  
    "shared_folder_id": "84528192421",  
    "to_dropbox_id": "dbid:AAEufNrMPSPe0dMQijRP0N_aZtBJRm26W4Q"  
}
```

TransferFolderArg

shared_folder_id *String(pattern="[-_0-9a-zA-Z:]")* The ID for the shared folder.

to_dropbox_id *String(min_length=1)* A account or team member ID to transfer ownership to.

RETURNS

No return values.

ERRORS Example: invalid_dropbox_id



≡

```
        ".tag": "invalid_dropbox_id"
    }
```

Example: new_owner_not_a_member

```
{
    "error_summary": "new_owner_not_a_member/...",
    "error": {
        ".tag": "new_owner_not_a_member"
    }
}
```

Example: new_owner_unmounted

```
{
    "error_summary": "new_owner_unmounted/...",
    "error": {
        ".tag": "new_owner_unmounted"
    }
}
```

Example: new_owner_email_unverified

```
{
    "error_summary": "new_owner_email_unverified/...",
    "error": {
        ".tag": "new_owner_email_unverified"
    }
}
```

Example: team_folder

```
{
    "error_summary": "team_folder/...",
    "error": {
        ".tag": "team_folder"
    }
}
```

Example: no_permission

```
{
    "error_summary": "no_permission/...",
    "error": {
        ".tag": "no_permission"
    }
}
```

Example: other

```
{
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

TransferFolderError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

access_error *SharedFolderAccessError*

invalid_dropbox_id *Void* TransferFolderArg.to_dropbox_id is invalid.

new_owner_not_a_member *Void* The new designated owner is not currently a member of the shared folder.

new_owner_unmounted *Void* The new designated owner has not added the folder to their Dropbox.

new_owner_email_unverified *Void* The new designated owner's e-mail address is unverified.

team_folder *Void* This action cannot be performed on a team shared folder.

no_permission *Void* The current user does not have permission to perform this action.

/unmount_folder

VERSION

1

DESCRIPTION

The current user unmounts the designated folder. They can re-mount the folder at a later time using [mount_folder](#).

This endpoint does not support apps with the app folder permission.



≡

X

AUTHENTICATION User Authentication, Dropbox-API-Select-Admin (Team Admin)

ENDPOINT FORMAT **RPC**

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/unmount_folder \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"shared_folder_id\": \"84528192421\"}"
```

PARAMETERS

```
{  
    "shared_folder_id": "84528192421"  
}
```

UnmountFolderArg**shared_folder_id** String(pattern="[-_0-9a-zA-Z:]") The ID for the shared folder.

RETURNS No return values.

ERRORS Example: no_permission

```
{  
    "error_summary": "no_permission/...",  
    "error": {  
        ".tag": "no_permission"  
    }  
}
```

Example: not_unmountable

```
{  
    "error_summary": "not_unmountable/...",  
    "error": {  
        ".tag": "not_unmountable"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

UnmountFolderError (open union)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

access_error SharedFolderAccessError**no_permission** Void The current user does not have permission to perform this action.**not_unmountable** Void The shared folder can't be unmounted. One example where this can occur is when the shared folder's parent folder is also a shared folder that resides in the current user's Dropbox.

/unshare_file

VERSION

1 ▾

DESCRIPTION Remove all members from this file. Does not remove inherited members.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

```
https://api.dropboxapi.com/2/sharing/unshare_file
```

AUTHENTICATION User Authentication, Dropbox-API-Select-Admin (Team Admin)

ENDPOINT FORMAT **RPC**

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/unshare_file \
--header "Authorization: Bearer <get access token>" \
```



PARAMETERS

```
{  
    "file": "id:3kmLmQFnf1AAAAAAAAAAw"  
}
```

UnshareFileArg

Arguments for `unshare_file`.

file `String(min_length=1, pattern="((/id:).*)|nspath:[0-9]+(:.*))|ns:[0-9]+(/.*)?")` The file to unshare.

RETURNS

No return values.

ERRORS

```
{  
    "error_summary": "user_error/email_unverified/...",  
    "error": {  
        ".tag": "user_error",  
        "user_error": {  
            ".tag": "email_unverified"  
        }  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

UnshareFileError (*open union*)

Error result for `unshare_file`. The value will be one of the following datatypes. New values may be introduced as our API evolves.

user_error `SharingUserError`

access_error `Sharing FileAccess Error`

/unshare_folder

VERSION

1 ▾

DESCRIPTION

Allows a shared folder owner to unshare the folder.

You'll need to call `check_job_status` to determine if the action has completed successfully.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

```
https://api.dropboxapi.com/2/sharing/unshare_folder
```

AUTHENTICATION

[User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/unshare_folder \  
--header "Authorization: Bearer <get access token>" \  
--header "Content-Type: application/json" \  
-data "{\"shared_folder_id\": \"84528192421\", \"leave_a_copy\": false}"
```

PARAMETERS

```
{  
    "shared_folder_id": "84528192421",  
    "leave_a_copy": false  
}
```

UnshareFolderArg

shared_folder_id `String(pattern="[-_0-9a-zA-Z:]")` The ID for the shared folder.

leave_a_copy `Boolean` If true, members of this shared folder will get a copy of this folder after it's unshared. Otherwise, it will be removed from their Dropbox. The current user, who is an owner, will always retain their copy. The default for this field is False.

RETURNS

Example: complete

```
{  
    ".tag": "complete"  
}
```



```
    ".tag": "async_job_id",
    "async_job_id": "34g93hh34h04y384084"
}
```

LaunchEmptyResult (*union*)

Result returned by methods that may either launch an asynchronous job or complete synchronously. Upon synchronous completion of the job, no additional information is returned. This datatype comes from an imported namespace originally defined in the `async` namespace. The value will be one of the following datatypes:

async_job_id *String(min_length=1)* This response indicates that the processing is asynchronous. The string is an id that can be used to obtain the status of the asynchronous job.

complete *Void* The job finished synchronously and successfully.

ERRORS Example: team_folder

```
{
  "error_summary": "team_folder/...",
  "error": {
    ".tag": "team_folder"
  }
}
```

Example: no_permission

```
{
  "error_summary": "no_permission/...",
  "error": {
    ".tag": "no_permission"
  }
}
```

Example: too_many_files

```
{
  "error_summary": "too_many_files/...",
  "error": {
    ".tag": "too_many_files"
  }
}
```

Example: other

```
{
  "error_summary": "other/...",
  "error": {
    ".tag": "other"
  }
}
```

UnshareFolderError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

access_error *SharedFolderAccessError*

team_folder *Void* This action cannot be performed on a team shared folder.

no_permission *Void* The current user does not have permission to perform this action.

too_many_files *Void* This shared folder has too many files to be unshared.

/update_file_member

VERSION

1

DESCRIPTION Changes a member's access on a shared file.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/update_file_member

AUTHENTICATION [User Authentication, Dropbox-API-Select-Admin \(Team Admin\)](#)

RPC

EXAMPLE Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/sharing/update_file_member \
--header "Authorization: Bearer <get access token>" \
```



☰ .tag\": \"email\", \"email\": \"justin@example.com\"

PARAMETERS

```
{  
  "file": "id:3kmLmQFnf1AAAAAAAAAAw",  
  "member": {  
    ".tag": "email",  
    "email": "justin@example.com"  
  },  
  "access_level": "viewer"  
}
```

UpdateFileMemberArgs

Arguments for `update_file_member`.

file `String(min_length=1, pattern="((/|id:).*)|nspath:[0-9]+(:*)|ns:[0-9]+(/.*)?)")` File for which we are changing a member's access.

member `MemberSelector` The member whose access we are changing.

access_level `AccessLevel` The new access level for the member.

RETURNS

MemberAccessLevelResult

Contains information about a member's access level to content after an operation.

access_level `AccessLevel?` The member still has this level of access to the content through a parent folder. This field is optional.

warning `String?` A localized string with additional information about why the user has this access level to the content. This field is optional.

access_details `List of (ParentFolderAccessInfo)?` The parent folders that a member has access to. The field is present if the user has access to the first parent folder where the member gains access. This field is optional.

ERRORS

Example: invalid_member

```
{  
  "error_summary": "invalid_member/...",  
  "error": {  
    ".tag": "invalid_member"  
  }  
}
```

Example: no_permission

```
{  
  "error_summary": "no_permission/...",  
  "error": {  
    ".tag": "no_permission"  
  }  
}
```

Example: other

```
{  
  "error_summary": "other/...",  
  "error": {  
    ".tag": "other"  
  }  
}
```

FileMemberActionError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

invalid_member `Void` Specified member was not found.

no_permission `Void` User does not have permission to perform this action on this member.

access_error `SharingFileAccessError` Specified file was invalid or user does not have access.

no_explicit_access `MemberAccessLevelResult` The action cannot be completed because the target member does not have explicit access to the file. The return value is the access that the member has to the file from a parent folder.

/update_folder_member

VERSION

1

DESCRIPTION Allows an owner or editor of a shared folder to update another member's permissions.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/update_folder_member

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/update_folder_member \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"shared_folder_id\": \"84528192421\", \"member\": {\".tag\": \"email\", \"email\": \"justin@example.com\"}}
```

PARAMETERS

```
{  
    \"shared_folder_id\": \"84528192421\",  
    \"member\": {  
        \".tag\": \"email\",  
        \"email\": \"justin@example.com\"  
    },  
    \"access_level\": \"editor\"\n}
```

UpdateFolderMemberArg

shared_folder_id *String(pattern="[-_0-9a-zA-Z:]")* The ID for the shared folder.

member *MemberSelector* The member of the shared folder to update. Only the MemberSelector.dropbox_id may be set at this time.

access_level *AccessLevel* The new access level for member. AccessLevel.owner is disallowed.

RETURNS

MemberAccessLevelResult

Contains information about a member's access level to content after an operation.

access_level *AccessLevel?* The member still has this level of access to the content through a parent folder. This field is optional.

warning *String?* A localized string with additional information about why the user has this access level to the content. This field is optional.

access_details *List of (ParentFolderAccessInfo)?* The parent folders that a member has access to. The field is present if the user has access to the first parent folder where the member gains access. This field is optional.

ERRORS

Example: insufficient_plan

```
{  
    \"error_summary\": \"insufficient_plan/...\",  
    \"error\": {  
        \".tag\": \"insufficient_plan\"\n    }\n}
```

Example: no_permission

```
{  
    \"error_summary\": \"no_permission/...\",  
    \"error\": {  
        \".tag\": \"no_permission\"\n    }\n}
```

Example: other

```
{  
    \"error_summary\": \"other/...\",  
    \"error\": {  
        \".tag\": \"other\"\n    }\n}
```

UpdateFolderMemberError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

access_error *SharedFolderAccessError*

member_error *SharedFolderMemberError*

no_explicit_access *AddFolderMemberError* If updating the access type required the member to be added to the shared folder and there was an error when adding the member.

insufficient_plan *Void* The current user's account doesn't support this action. An example of this is when downgrading a member from editor to viewer. This action can only be performed by users that have upgraded to a Pro or Business plan.

no_permission *Void* The current user does not have permission to perform this action.



Version: 1 ▾

DESCRIPTION Update the sharing policies for a shared folder.
User must have AccessLevel.owner access to the shared folder to update its policies.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE https://api.dropboxapi.com/2/sharing/update_folder_policy

AUTHENTICATION [User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT FORMAT [RPC](#)

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/update_folder_policy \
  --header "Authorization: Bearer <get access token>" \
  --header "Content-Type: application/json" \
  --data "{\"shared_folder_id\": \"84528192421\", \"member_policy\": \"team\", \"acl_update_policy\": \"owner\", \"shared_link_policy\": \"members\"}
```

PARAMETERS

```
{\n    \"shared_folder_id\": \"84528192421\",\n    \"member_policy\": \"team\",\n    \"acl_update_policy\": \"owner\",\n    \"shared_link_policy\": \"members\"\n}
```

UpdateFolderPolicyArg

If any of the policies are unset, then they retain their current setting.

shared_folder_id *String(pattern="[-_0-9a-zA-Z:]")* The ID for the shared folder.

member_policy *MemberPolicy?* Who can be a member of this shared folder. Only applicable if the current user is on a team. This field is optional.

acl_update_policy *AclUpdatePolicy?* Who can add and remove members of this shared folder. This field is optional.

viewer_info_policy *ViewerInfoPolicy?* Who can enable/disable viewer info for this shared folder. This field is optional.

shared_link_policy *SharedLinkPolicy?* The policy to apply to shared links created for content inside this shared folder. The current user must be on a team to set this policy to SharedLinkPolicy.members. This field is optional.

link_settings *LinkSettings?* Settings on the link for this folder. This field is optional.

actions *List of (FolderAction)?* A list of `FolderAction`'s corresponding to `FolderPermission`'s that should appear in the response's SharedFolderMetadata.permissions field describing the actions the authenticated user can perform on the folder. This field is optional.

RETURNS

```
{\n    \"access_type\": {\n        \".tag\": \"owner\"\n    },\n    \"is_inside_team_folder\": false,\n    \"is_team_folder\": false,\n    \"name\": \"dir\",\n    \"policy\": {\n        \"acl_update_policy\": {\n            \".tag\": \"owner\"\n        },\n        \"shared_link_policy\": {\n            \".tag\": \"anyone\"\n        },\n        \"member_policy\": {\n            \".tag\": \"anyone\"\n        },\n        \"resolved_member_policy\": {\n            \".tag\": \"team\"\n        }\n    },\n    \"preview_url\": \"https://www.dropbox.com/scl/fo/fir9vjelf\", \n    \"shared_folder_id\": \"84528192421\", \n    \"time_invited\": \"2016-01-20T00:00:00Z\", \n    \"path_lower\": \"/dir\", \n    \"link_metadata\": {\n        \"audience_options\": [\n            {\n                \".tag\": \"public\"\n            },\n            {\n                \".tag\": \"team\"\n            }\n        ]\n    }\n}
```



```
        ".tag": "public"
    },
    "link_permissions": [
        {
            "action": {
                ".tag": "change_audience"
            },
            "allow": true
        }
    ],
    "password_protected": false,
    "url": ""
},
"permissions": [],
"access_inheritance": {
    ".tag": "inherit"
}
}
```

SharedFolderMetadata

The metadata which includes basic information about the shared folder.

access_type *AccessLevel* The current user's access level for this shared folder.

is_inside_team_folder *Boolean* Whether this folder is inside of a team folder.

is_team_folder *Boolean* Whether this folder is a [team folder](#).

name *String* The name of the this shared folder.

policy *FolderPolicy* Policies governing this shared folder.

preview_url *String* URL for displaying a web preview of the shared folder.

shared_folder_id *String(pattern="[-_0-9a-zA-Z:]")* The ID of the shared folder.

time_invited *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")* Timestamp indicating when the current user was invited to this shared folder.

owner_display_names *List of (String)?* The display names of the users that own the folder. If the folder is part of a team folder, the display names of the team admins are also included. Absent if the owner display names cannot be fetched. This field is optional.

owner_team *Team?* The team that owns the folder. This field is not present if the folder is not owned by a team. This field is optional.

parent_shared_folder_id *String(pattern="[-_0-9a-zA-Z:]")?* The ID of the parent shared folder. This field is present only if the folder is contained within another shared folder. This field is optional.

path_lower *String?* The lower-cased full path of this shared folder. Absent for unmounted folders. This field is optional.

parent_folder_name *String?* Display name for the parent folder. This field is optional.

link_metadata *SharedContentLinkMetadata?* The metadata of the shared content link to this shared folder. Absent if there is no link on the folder. This is for an unreleased feature so it may not be returned yet. This field is optional.

permissions *List of (FolderPermission)?* Actions the current user may perform on the folder and its contents. The set of permissions corresponds to the FolderActions in the request. This field is optional.

access_inheritance *AccessInheritance* Whether the folder inherits its members from its parent. The default for this union is inherit.

ERRORS

Example: not_on_team

```
{
    "error_summary": "not_on_team/...",
    "error": {
        ".tag": "not_on_team"
    }
}
```

Example: team_policy_disallows_member_policy

```
{
    "error_summary": "team_policy_disallows_member_policy/...",
    "error": {
        ".tag": "team_policy_disallows_member_policy"
    }
}
```

Example: disallowed_shared_link_policy

```
{
    "error_summary": "disallowed_shared_link_policy/...",
    "error": {
```



≡

Example: no_permission

```
{  
    "error_summary": "no_permission/...",  
    "error": {  
        ".tag": "no_permission"  
    }  
}
```

Example: team_folder

```
{  
    "error_summary": "team_folder/...",  
    "error": {  
        ".tag": "team_folder"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

UpdateFolderPolicyError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

access_error *SharedFolderAccessError*

not_on_team *Void* `UpdateFolderPolicyArg.member_policy` was set even though user is not on a team.

team_policy_disallows_member_policy *Void* Team policy is more restrictive than `ShareFolderArg.member_policy`.

disallowed_shared_link_policy *Void* The current account is not allowed to select the specified `ShareFolderArg.shared_link_policy`.

no_permission *Void* The current user does not have permission to perform this action.

team_folder *Void* This action cannot be performed on a team shared folder.

team

team_log

users

This namespace contains endpoints and data types for user management.

/features/get_values

VERSION

DESCRIPTION Get a list of feature values that may be configured for the current account.

URL STRUCTURE

https://api.dropboxapi.com/2/users/features/get_values

AUTHENTICATION [User Authentication](#)

ENDPOINT FORMAT

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/users/features/get_values \  
--header "Authorization: Bearer <get access token>" \  
--header "Content-Type: application/json" \  
--data "{\"features\": [{\".tag\": \"paper_as_files\"}, {\".tag\": \"file_locking\"}]}"
```

PARAMETERS

```
{  
    "features": [  
        {  
            ".tag": "paper_as_files"  
        },  
        {  
            ".tag": "file_locking"  
        }  
    ]  
}
```




Example: team

```
{  
    "account_id": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc",  
    "name": {  
        "given_name": "Franz",  
        "surname": "Ferdinand",  
        "familiar_name": "Franz",  
        "display_name": "Franz Ferdinand (Personal)",  
        "abbreviated_name": "FF"  
    },  
    "email": "franz@dropbox.com",  
    "email_verified": true,  
    "disabled": false,  
    "is_teammate": true,  
    "profile_photo_url": "https://dl-web.dropbox.com/account_photo/get/dbaphid%3AAAHWGMIXV3sUu0mBfTz0wPsiqHUpBWvv3z  
    "team_member_id": "dbmid:AAHy7WsR0x-u4ZCqiDl5Fz5zvuL3kmspU"  
}
```

BasicAccount

Basic information about any account.

account_id *String(min_length=40, max_length=40)* The user's unique Dropbox ID.

name *Name* Details of a user's name.

email *String* The user's e-mail address. Do not rely on this without checking the email_verified field. Even then, it's possible that the user has since lost access to their e-mail.

email_verified *Boolean* Whether the user has verified their e-mail address.

disabled *Boolean* Whether the user has been disabled.

is_teammate *Boolean* Whether this user is a teammate of the current user. If this account is the current user's account, then this will be true.

profile_photo_url *String?* URL for the photo representing the user, if one is set. This field is optional.

team_member_id *String?* The user's unique team member id. This field will only be present if the user is part of a team and is_teammate is true. This field is optional.

ERRORS

Example: no_account

```
{  
    "error_summary": "no_account/...",  
    "error": {  
        ".tag": "no_account"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

GetAccountError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

no_account *Void* The specified GetAccountArg.account_id does not exist.

/get_account_batch

VERSION

1

DESCRIPTION Get information about multiple user accounts. At most 300 accounts may be queried per request.

URL STRUCTURE

https://api.dropboxapi.com/2/users/get_account_batch

AUTHENTICATION

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot





\

```
--data "{\"account_ids\": [\"dbid:AAH4f99T0taONib-OurWxbNQ6ywGRopQngc\", \"dbid:AAH1Vcz-DVoRDeixtr_OA8oUGgiqhs4\"]}
```

PARAMETERS

```
{  
  "account_ids": [  
    "dbid:AAH4f99T0taONib-OurWxbNQ6ywGRopQngc",  
    "dbid:AAH1Vcz-DVoRDeixtr_OA8oUGgiqhs4XP0Q"  
  ]  
}
```

GetAccountBatchArg

account_ids *List of (String(min_length=40, max_length=40), min_items=1)* List of user account identifiers. Should not contain any duplicate account IDs.

RETURNS This route returns a list. This means the route can accept a homogenous list of the following types:

BasicAccount

Basic information about any account.

account_id *String(min_length=40, max_length=40)* The user's unique Dropbox ID.

name *Name* Details of a user's name.

email *String* The user's e-mail address. Do not rely on this without checking the email_verified field. Even then, it's possible that the user has since lost access to their e-mail.

email_verified *Boolean* Whether the user has verified their e-mail address.

disabled *Boolean* Whether the user has been disabled.

is_teammate *Boolean* Whether this user is a teammate of the current user. If this account is the current user's account, then this will be true.

profile_photo_url *String?* URL for the photo representing the user, if one is set. This field is optional.

team_member_id *String?* The user's unique team member id. This field will only be present if the user is part of a team and is_teammate is true. This field is optional.

ERRORS

```
{  
  "error_summary": "no_account/...",  
  "error": {  
    ".tag": "no_account",  
    "no_account": "dbid:AAH4f99T0taONib-OurWxbNQ6ywGRopQngc"  
  }  
}
```

Example: other

```
{  
  "error_summary": "other/...",  
  "error": {  
    ".tag": "other"  
  }  
}
```

GetAccountBatchError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

no_account *String(min_length=40, max_length=40)* The value is an account ID specified in GetAccountBatchArg.account_ids that does not exist.

/get_current_account

VERSION

1



DESCRIPTION

Get information about the current user's account.

URL STRUCTURE

https://api.dropboxapi.com/2/users/get_current_account

AUTHENTICATION

[User Authentication, Dropbox-API-Select-Admin \(Whole Team\)](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot



```
curl -X POST https://api.dropboxapi.com/2/users/get_current_account \  
-header "Authorization: Bearer <get access token>"
```



RETURNS Example: Paired account.

```
{  
    "account_id": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc",  
    "name": {  
        "given_name": "Franz",  
        "surname": "Ferdinand",  
        "familiar_name": "Franz",  
        "display_name": "Franz Ferdinand (Personal)",  
        "abbreviated_name": "FF"  
    },  
    "email": "franz@dropbox.com",  
    "email_verified": true,  
    "disabled": false,  
    "locale": "en",  
    "referral_link": "https://db.tt/ZITNuhtI",  
    "is_paired": true,  
    "account_type": {  
        ".tag": "business"  
    },  
    "root_info": {  
        ".tag": "user",  
        "root_namespace_id": "3235641",  
        "home_namespace_id": "3235641"  
    },  
    "country": "US",  
    "team": {  
        "id": "dbtid:AAFdgehTzw7WlXhZJsbGCLePe8RvQQYDr-I",  
        "name": "Acme, Inc.",  
        "sharing_policies": {  
            "shared_folder_member_policy": {  
                ".tag": "team"  
            },  
            "shared_folder_join_policy": {  
                ".tag": "from_anyone"  
            },  
            "shared_link_create_policy": {  
                ".tag": "team_only"  
            }  
        },  
        "office_addin_policy": {  
            ".tag": "disabled"  
        }  
    },  
    "team_member_id": "dbmid:AAHy7WsR0x-u4ZCqiDl5Fz5zvuL3kmfspwU"  
}
```

Example: A personal account that is not paired with a team.

```
{  
    "account_id": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc",  
    "name": {  
        "given_name": "Franz",  
        "surname": "Ferdinand",  
        "familiar_name": "Franz",  
        "display_name": "Franz Ferdinand (Personal)",  
        "abbreviated_name": "FF"  
    },  
    "email": "franz@gmail.com",  
    "email_verified": false,  
    "disabled": false,  
    "locale": "en",  
    "referral_link": "https://db.tt/ZITNuhtI",  
    "is_paired": false,  
    "account_type": {  
        ".tag": "basic"  
    },  
    "root_info": {  
        ".tag": "user",  
        "root_namespace_id": "3235641",  
        "home_namespace_id": "3235641"  
    },  
    "profile_photo_url": "https://dl-web.dropbox.com/account_photo/get/dbaphid%3AAAHWGmIXV3sUu0mBfTz0wPsiqHUpBWvv3z",  
    "country": "US"  
}
```

FullAccount

Detailed information about the current user's account.

account_id *String(min_length=40, max_length=40)* The user's unique Dropbox ID.

name *Name* Details of a user's name.

email *String* The user's e-mail address. Do not rely on this without checking the email_verified field. Even then, it's possible that the user has since lost access to their e-mail.

email_verified *Boolean* Whether the user has verified their e-mail address.

disabled *Boolean* Whether the user has been disabled.

locale *String(min_length=2)* The language that the user specified. Locale tags will be [IETF language tags](#).

referral_link *String* The user's referral link.

is_paired *Boolean* Whether the user has a personal and work account. If the current account is personal, then team will always be None, but is_paired will indicate if a work account is linked.

account_type *AccountType* What type of account this user has.



country *String(min_length=2, max_length=2)?* The user's two-letter country code, if available. Country codes are based on [ISO 3166-1](#). This field is optional.

team *FullTeam?* If this account is a member of a team, information about that team. This field is optional.

team_member_id *String?* This account's unique team member id. This field will only be present if team is present. This field is optional.

ERRORS No errors.

/get_space_usage

VERSION 1 ▾

DESCRIPTION Get the space usage information for the current user's account.

URL STRUCTURE https://api.dropboxapi.com/2/users/get_space_usage

AUTHENTICATION [User Authentication](#)

RPC

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/users/get_space_usage \
      --header "Authorization: Bearer <get access token>"
```

PARAMETERS No parameters.

RETURNS

```
{  
    "used": 314159265,  
    "allocation": {  
        ".tag": "individual",  
        "allocated": 10000000000  
    }  
}
```

SpaceUsage

Information about a user's space usage and quota.

used *UInt64* The user's total space usage (bytes).

allocation *SpaceAllocation* The user's space allocation.

ERRORS No errors.

deprecated

All deprecated endpoints, across namespaces

/alpha/get_metadata

VERSION 1 ▾

DEPRECATED BY /get_metadata

PREVIEW - may change or disappear without notice

DESCRIPTION Returns the metadata for a file or folder. This is an alpha endpoint compatible with the properties API.

Note: Metadata for the root folder is unsupported.

URL STRUCTURE https://api.dropboxapi.com/2/files/alpha/get_metadata

AUTHENTICATION [User Authentication](#)

RPC

EXAMPLE Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/alpha/get_metadata \
      --header "Authorization: Bearer <get access token>" \
```



≡

false, \\"include_deleted\": false, \\"include_has_ex

PARAMETERS

```
{  
    "path": "/Homework/math",  
    "include_media_info": false,  
    "include_deleted": false,  
    "include_has_explicit_shared_members": false  
}
```

Example: id

```
{  
    "path": "id:a4ayc_80_0EAAAAAAAAYa",  
    "include_media_info": false,  
    "include_deleted": false,  
    "include_has_explicit_shared_members": false  
}
```

Example: rev

```
{  
    "path": "rev:alc10ce0dd78",  
    "include_media_info": false,  
    "include_deleted": false,  
    "include_has_explicit_shared_members": false  
}
```

AlphaGetMetadataArg

path *String(pattern="/(.|[\\r\\n])*|id:.*|(rev:[0-9a-f]{9,})|(ns:[0-9]+(.*)?)"")* The path of a file or folder on Dropbox.

include_media_info *Boolean* If true, FileMetadata.media_info is set for photo and video. The default for this field is False.

include_deleted *Boolean* If true, DeletedMetadata will be returned for deleted file or folder, otherwise LookupError.not_found will be returned. The default for this field is False.

include_has_explicit_shared_members *Boolean* If true, the results will include a flag for each file indicating whether or not that file has any explicit members. The default for this field is False.

include_property_groups *TemplateFilterBase?* If set to a valid list of template IDs, FileMetadata.property_groups is set if there exists property data associated with the file and each of the listed templates. This field is optional.

include_property_templates *[deprecated] List of (String(min_length=1, pattern="/|ptid:.*"))?* Field is deprecated. If set to a valid list of template IDs, FileMetadata.property_groups is set for files with custom properties. This field is optional.

RETURNS

```
{  
    ".tag": "file",  
    "name": "Prime_Numbers.txt",  
    "id": "id:a4ayc_80_0EAAAAAAAAXw",  
    "client_modified": "2015-05-12T15:50:38Z",  
    "server_modified": "2015-05-12T15:50:38Z",  
    "rev": "alc10ce0dd78",  
    "size": 7212,  
    "path_lower": "/homework/math/prime_numbers.txt",  
    "path_display": "/Homework/math/Prime_Numbers.txt",  
    "sharing_info": {  
        "read_only": true,  
        "parent_shared_folder_id": "84528192421",  
        "modified_by": "dbid:AAH4f99T0ta0N1b-OurWxbNQ6ywGRopQngc"  
    },  
    "is_downloadable": true,  
    "property_groups": [  
        {  
            "template_id": "ptid:1a5n2i6d30YEAAAAAAAAYa",  
            "fields": [  
                {  
                    "name": "Security Policy",  
                    "value": "Confidential"  
                }  
            ]  
        }  
    ],  
    "has_explicit_shared_members": false,  
    "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",  
    "file_lock_info": {  
        "is_lockholder": true,  
        "lockholder_name": "Imaginary User",  
        "created": "2015-05-12T15:50:38Z"  
    }  
}
```

```
{  
    ".tag": "folder",  
    "name": "math",  
    "id": "id:a4ayc_80_0EAAAAAAAAXz",  
    "path_lower": "/homework/math",  
    "path_display": "/Homework/math",  
    "sharing_info": {  
        "read_only": false,  
        "parent_shared_folder_id": "84528192421",  
        "traverse_only": false,  
    }
```



```
"template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",
"fields": [
    {
        "name": "Security Policy",
        "value": "Confidential"
    }
]
```

Metadata (*datatype with subtypes*)

Metadata for a file or folder. This datatype will be one of the following subtypes:

file *FileMetadata*

folder *FolderMetadata*

deleted *DeletedMetadata*

ERRORS

AlphaGetMetadataError (*union*)

The value will be one of the following datatypes:

path *LookupError*

properties_error *LookUpPropertiesError*

/alpha/upload

VERSION

1

DEPRECATED BY version 1

PREVIEW - may change or disappear without notice

DESCRIPTION

Create a new file with the contents provided in the request. Note that this endpoint is part of the properties API alpha and is slightly different from [upload](#).

Do not use this to upload a file larger than 150 MB. Instead, create an upload session with [upload_session/start](#).

URL STRUCTURE

<https://content.dropboxapi.com/2/files/alpha/upload>

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[Content-upload](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://content.dropboxapi.com/2/files/alpha/upload \
--header "Authorization: Bearer <get access token>" \
--header "Dropbox-API-Arg: {\"path\": \"/Homework/math/Matrices.txt\", \"mode\": \"add\", \"autorename\": true, \
--header \"Content-Type: application/octet-stream\" \
--data-binary @local_file.txt
```

PARAMETERS

```
{
    "path": "/Homework/math/Matrices.txt",
    "mode": "add",
    "autorename": true,
    "mute": false,
    "strict_conflict": false
}
```

CommitInfoWithProperties

path *String(pattern="/(.|[\\r\\n])*|(ns:[0-9]+(?:\\.*))|(id:*)")* Path in the user's Dropbox to save the file.

mode *WriteMode* Selects what to do if the file already exists. The default for this union is add.

autorename *Boolean* If there's a conflict, as determined by mode, have the Dropbox server try to autorename the file to avoid conflict. The default for this field is False.

client_modified *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")?* The value to store as the client_modified timestamp. Dropbox automatically records the time at which the file was written to the Dropbox servers. It can also record an additional timestamp, provided by Dropbox desktop clients, mobile clients, and API apps of when the file was actually created or modified. This field is optional.

mute *Boolean* Normally, users are made aware of any file modifications in their Dropbox account via notifications in the client software. If true, this tells the clients that this modification shouldn't result in a user notification. The default for this field is False.

property_groups *List of (PropertyGroup)?* List of custom properties to add to file. This field is optional.



match the existing file's "rev", even if the existing file has been deleted. The default for this field is False.

RETURNS

```
{  
    "name": "Prime_Numbers.txt",  
    "id": "id:aayc_80_0EAAAAAAAXw",  
    "client_modified": "2015-05-12T15:50:38Z",  
    "server_modified": "2015-05-12T15:50:38Z",  
    "rev": "alc10ce0dd78",  
    "size": 7212,  
    "path_lower": "/homework/math/prime_numbers.txt",  
    "path_display": "/Homework/math/Prime_Numbers.txt",  
    "sharing_info": {  
        "read_only": true,  
        "parent_shared_folder_id": "84528192421",  
        "modified_by": "dbid:ÂAH4f99T0taONIb-OurWxbNQ6ywGRopQngc"  
    },  
    "is_downloadable": true,  
    "property_groups": [  
        {  
            "template_id": "ptid:la5n2i6d30YEAAAAAAAAYa",  
            "fields": [  
                {  
                    "name": "Security Policy",  
                    "value": "Confidential"  
                }  
            ]  
        }  
    ],  
    "has_explicit_shared_members": false,  
    "content_hash": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",  
    "file_lock_info": {  
        "is_lockholder": true,  
        "lockholder_name": "Imaginary User",  
        "created": "2015-05-12T15:50:38Z"  
    }  
}
```

FileMetadata

name *String* The last component of the path (including extension). This never contains a slash.

id *String(min_length=1)* A unique identifier for the file.

client_modified *Timestamp(format="%Y-%m-%dT%H:%M:%S")*

For files, this is the modification time set by the desktop client when the file was added to Dropbox. Since this time is not verified (the Dropbox server stores whatever the desktop client sends up), this should only be used for display purposes (such as sorting) and not, for example, to determine if a file has changed or not.

server_modified *Timestamp(format="%Y-%m-%dT%H:%M:%S")*
The last time the file was modified on Dropbox.

rev *String(min_length=9, pattern="[0-9a-f]+")* A unique identifier for the current revision of a file. This field is the same rev as elsewhere in the API and can be used to detect changes and avoid conflicts.

size *UInt64* The file size in bytes.

path_lower *String?* The lowercased full path in the user's Dropbox. This always starts with a slash. This field will be null if the file or folder is not mounted. This field is optional.

path_display *String?* The cased path to be used for display purposes only. In rare instances the casing will not correctly match the user's filesystem, but this behavior will match the path provided in the Core API v1, and at least the last path component will have the correct casing. Changes to only the casing of paths won't be returned by [list_folder/continue](#). This field will be null if the file or folder is not mounted. This field is optional.

parent_shared_folder_id *(deprecated) String(pattern="[-_0-9a-zA-Z:j+)")*? Field is deprecated. Please use FileSharingInfo.parent_shared_folder_id or FolderSharingInfo.parent_shared_folder_id instead. This field is optional.

media_info *MediaInfo?* Additional information if the file is a photo or video. This field will not be set on entries returned by [list_folder](#), [list_folder/continue](#), or [get_thumbnail_batch](#), starting December 2, 2019. This field is optional.

symlink_info *SymlinkInfo?* Set if this file is a symlink. This field is optional.

sharing_info *FileSharingInfo?* Set if this file is contained in a shared folder. This field is optional.

is_downloadable *Boolean* If true, file can be downloaded directly; else the file must be exported. The default for this field is True.

export_info *ExportInfo?* Information about format this file can be exported to. This field must be set if is_downloadable is set to false. This field is optional.



has_explicit_shared_members Boolean? This flag will only be present if include_has_explicit_shared_members is true in [list_folder](#) or [get_metadata](#). If this flag is present, it will be true if this file has any explicit shared members. This is different from sharing_info in that this could be true in the case where a file has explicit members but is not contained within a shared folder. This field is optional.

content_hash String(min_length=64, max_length=64)? A hash of the file content. This field can be used to verify data integrity. For more information see our [Content hash](#) page. This field is optional.

file_lock_info FileLockMetadata? If present, the metadata associated with the file's current lock. This field is optional.

ERRORS

```
{  
    "error_summary": "properties_error/does_not_fit_template/...",  
    "error": {  
        ".tag": "properties_error",  
        "properties_error": {  
            ".tag": "does_not_fit_template"  
        }  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

UploadErrorWithProperties (union)

The value will be one of the following datatypes:

path UploadWriteFailed Unable to save the uploaded contents to a file.

properties_error InvalidPropertyGroupError The supplied property group is invalid. The file has uploaded without property groups.

/properties/add

VERSION

1 ▾

DEPRECATED

DESCRIPTION

No description

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

<https://api.dropboxapi.com/2/files/properties/add>

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/properties/add \  
    --header "Authorization: Bearer <get access token>" \  
    --header "Content-Type: application/json" \  
    --data "{\"path\": \"/my_awesome/word.docx\", \"property_groups\": [{\"template_id\": \"ptid:1a5n2i6d30YEAAAAAAAAYa\"}]}
```

PARAMETERS

```
{  
    "path": "/my_awesome/word.docx",  
    "property_groups": [  
        {  
            "template_id": "ptid:1a5n2i6d30YEAAAAAAAAYa",  
            "fields": [  
                {  
                    "name": "Security Policy",  
                    "value": "Confidential"  
                }  
            ]  
        }  
    ]  
}
```

AddPropertiesArg

This datatype comes from an imported namespace originally defined in the `file_properties` namespace.



which are to be added to a Dropbox file.

RETURNS No return values.

EXAMPLES Example: restricted_content

```
{  
    "error_summary": "restricted_content/...",  
    "error": {  
        ".tag": "restricted_content"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: unsupported_folder

```
{  
    "error_summary": "unsupported_folder/...",  
    "error": {  
        ".tag": "unsupported_folder"  
    }  
}
```

Example: property_field_too_large

```
{  
    "error_summary": "property_field_too_large/...",  
    "error": {  
        ".tag": "property_field_too_large"  
    }  
}
```

Example: does_not_fit_template

```
{  
    "error_summary": "does_not_fit_template/...",  
    "error": {  
        ".tag": "does_not_fit_template"  
    }  
}
```

Example: property_group_already_exists

```
{  
    "error_summary": "property_group_already_exists/...",  
    "error": {  
        ".tag": "property_group_already_exists"  
    }  
}
```

AddPropertiesError (*union*)

This datatype comes from an imported namespace originally defined in the file_properties namespace. The value will be one of the following datatypes:

template_not_found *String*(min_length=1, pattern="(\\|ptid:).*)")
Template does not exist for the given identifier.

restricted_content *Void* You do not have permission to modify this template.

path *LookupError*

unsupported_folder *Void* This folder cannot be tagged. Tagging folders is not supported for team-owned templates.

property_field_too_large *Void* One or more of the supplied property field values is too large.

does_not_fit_template *Void* One or more of the supplied property fields does not conform to the template specifications.

property_group_already_exists *Void* A property group associated with this template and file already exists.

/properties/overwrite

VERSION

1

DEPRECATED

DESCRIPTION No description



URL STRUCTURE

`https://api.dropboxapi.com/2/files/properties/overwrite`

AUTHENTICATION

User Authentication

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/files/properties/overwrite \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"path\": \"/my_awesome/word.docx\", \"property_groups\": [{\"template_id\": \"ptid:1a5n2i6d30YEAAAAAAA\"}]}
```

PARAMETERS

```
{  
    "path": "/my_awesome/word.docx",  
    "property_groups": [  
        {  
            "template_id": "ptid:1a5n2i6d30YEAAAAAAAAYa",  
            "fields": [  
                {  
                    "name": "Security Policy",  
                    "value": "Confidential"  
                }  
            ]  
        }  
    ]  
}
```

OverwritePropertyGroupArg

This datatype comes from an imported namespace originally defined in the file_properties namespace.

path `String(pattern="/(.|[\\r\\n])*|id:.*|(ns:[0-9]+(/.*)?)"")` A unique identifier for the file or folder.

property_groups `List of (PropertyGroup, min_items=1)` The property groups "snapshot" updates to force apply.

RETURNS

No return values.

ERRORS

Example: restricted_content

```
{  
    "error_summary": "restricted_content/...",  
    "error": {  
        ".tag": "restricted_content"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: unsupported_folder

```
{  
    "error_summary": "unsupported_folder/...",  
    "error": {  
        ".tag": "unsupported_folder"  
    }  
}
```

Example: property_field_too_large

```
{  
    "error_summary": "property_field_too_large/...",  
    "error": {  
        ".tag": "property_field_too_large"  
    }  
}
```

Example: does_not_fit_template

```
{  
    "error_summary": "does_not_fit_template/...",  
    "error": {  
        ".tag": "does_not_fit_template"  
    }  
}
```

InvalidPropertyGroupError (*union*)



ned



template_not_found *String(min_length=1, pattern="/|ptid:).*)")*
Template does not exist for the given identifier.

restricted_content *Void* You do not have permission to modify this template.

path *LookupError*

unsupported_folder *Void* This folder cannot be tagged. Tagging folders is not supported for team-owned templates.

property_field_too_large *Void* One or more of the supplied property field values is too large.

does_not_fit_template *Void* One or more of the supplied property fields does not conform to the template specifications.

/properties/remove

VERSION

1

DEPRECATED

DESCRIPTION

No description

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

`https://api.dropboxapi.com/2/files/properties/remove`

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/files/properties/remove \
    --header "Authorization: Bearer <get access token>" \
    --header "Content-Type: application/json" \
    --data "{\"path\": \"/my_awesome/word.docx\", \"property_template_ids\": [\"ptid:la5n2i6d30YEAFFFFFFAYa\"]}"
```

PARAMETERS

```
{  
    "path": "/my_awesome/word.docx",  
    "property_template_ids": [  
        "ptid:la5n2i6d30YEAFFFFFFAYa"  
    ]  
}
```

RemovePropertiesArg

This datatype comes from an imported namespace originally defined in the file_properties namespace.

path *String(pattern="/(.|[r\n])*|id:.*|(ns:[0-9]+(/.*)?))")* A unique identifier for the file or folder.

property_template_ids *List of (String(min_length=1, pattern="/|ptid:).*)")* A list of identifiers for a template created by [templates/add_for_user](#) or [templates/add_for_team](#).

RETURNS

No return values.

ERRORS

Example: restricted_content

```
{  
    "error_summary": "restricted_content/...",  
    "error": {  
        ".tag": "restricted_content"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: unsupported_folder

```
{  
    "error_summary": "unsupported_folder/...",  
    "error": {  
        ".tag": "unsupported_folder"  
    }  
}
```



≡



removePropertiesError (union)

This datatype comes from an imported namespace originally defined in the file_properties namespace. The value will be one of the following datatypes:

template_not_found *String(min_length=1, pattern="(\\|ptid:).*)")*

Template does not exist for the given identifier.

restricted_content *Void* You do not have permission to modify this template.

path *LookupError*

unsupported_folder *Void* This folder cannot be tagged. Tagging folders is not supported for team-owned templates.

property_group_lookup *LookUpPropertiesError*

/properties/template/get

VERSION

1

DEPRECATED

DESCRIPTION

No description

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

<https://api.dropboxapi.com/2/files/properties/template/get>

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/files/properties/template/get \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"template_id\": \"ptid:la5n2i6d30YEAAAAAAAAYa\"}"
```

PARAMETERS

```
{\n    \"template_id\": \"ptid:la5n2i6d30YEAAAAAAAAYa\"\n}
```

GetTemplateArg

This datatype comes from an imported namespace originally defined in the file_properties namespace.

template_id *String(min_length=1, pattern="(\\|ptid:).*)")* An identifier for template added by route See [templates/add_for_user](#) or [templates/add_for_team](#).

RETURNS

```
{\n    \"name\": \"Security\",\n    \"description\": \"These properties describe how confidential this file or folder is.\",\n    \"fields\": [\n        {\n            \"name\": \"Security Policy\",\n            \"description\": \"This is the security policy of the file or folder described.\n\nPolicies can be Confidential, Sensitive, or Standard.\",\n            \"type\": {\n                \".tag\": \"string\"\n            }\n        }\n    ]\n}
```

GetTemplateResult

This datatype comes from an imported namespace originally defined in the file_properties namespace.

name *String* Display name for the template. Template names can be up to 256 bytes.

description *String* Description for the template. Template descriptions can be up to 1024 bytes.

fields *List of (PropertyFieldTemplate)* Definitions of the property fields associated with this template. There can be up to 32 properties in a single template.

ERRORS Example: restricted_content

```
{\n    \"error_summary\": \"restricted_content/...\", \n    \"error\": {\n        \".tag\": \"restricted_content\"\n    }\n}
```



≡



Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

TemplateError (*open union*)

This datatype comes from an imported namespace originally defined in the file_properties namespace. The value will be one of the following datatypes. New values may be introduced as our API evolves.

template_not_found *String(min_length=1, pattern="(\\|ptid:).*)")*
Template does not exist for the given identifier.

restricted_content *Void* You do not have permission to modify this template.

/properties/template/list

VERSION

1

DEPRECATED

DESCRIPTION

No description

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

<https://api.dropboxapi.com/2/files/properties/template/list>

AUTHENTICATION

User Authentication

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/files/properties/template/list \  
--header "Authorization: Bearer <get access token>"
```

PARAMETERS

No parameters.

RETURNS

```
{  
    "template_ids": [  
        "ptid:la5n2i6d30YEAFFFFFFAYa"  
    ]  
}
```

ListTemplateResult

This datatype comes from an imported namespace originally defined in the file_properties namespace.

template_ids *List of (String(min_length=1, pattern="(\\|ptid:).*)")*
List of identifiers for templates added by See [templates/add_for_user](#) or [templates/add_for_team](#).

ERRORS

Example: restricted_content

```
{  
    "error_summary": "restricted_content/...",  
    "error": {  
        ".tag": "restricted_content"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

TemplateError (*open union*)

This datatype comes from an imported namespace originally defined in the file_properties namespace. The value will be one of the following datatypes. New values may be introduced as our API evolves.



this template.

/properties/update

VERSION

1 ▾

DEPRECATED

DESCRIPTION

No description

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

<https://api.dropboxapi.com/2/files/properties/update>

AUTHENTICATION

User Authentication

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/files/properties/update \
      --header "Authorization: Bearer <get access token>" \
      --header "Content-Type: application/json" \
      --data "{\"path\": \"/my_awesome/word.docx\", \"update_property_groups\": [{\"template_id\": \"ptid:la5n2i6d30YF\", \"add_or_update_fields\": [ {\"name\": \"Security Policy\", \"value\": \"Confidential\" } ], \"remove_fields\": []}]}"
```

PARAMETERS

```
{ "path": "/my_awesome/word.docx",
  "update_property_groups": [
    {
      "template_id": "ptid:la5n2i6d30YF",
      "add_or_update_fields": [
        {
          "name": "Security Policy",
          "value": "Confidential"
        }
      ],
      "remove_fields": []
    }
  ]
}
```

UpdatePropertiesArg

This datatype comes from an imported namespace originally defined in the file_properties namespace.

path *String(pattern="/(.|[\\r\\n])*|id:.*|(ns:[0-9]+(/.*)?))* A unique identifier for the file or folder.

update_property_groups *List of (PropertyGroupUpdate)* The property groups "delta" updates to apply.

RETURNS

No return values.

ERRORS

Example: restricted_content

```
{ "error_summary": "restricted_content/...",
  "error": {
    ".tag": "restricted_content"
  }
}
```

Example: other

```
{ "error_summary": "other/...",
  "error": {
    ".tag": "other"
  }
}
```

Example: unsupported_folder

```
{ "error_summary": "unsupported_folder/...",
  "error": {
    ".tag": "unsupported_folder"
  }
}
```

Example: property_field_too_large

```
{ "error_summary": "property_field_too_large/...",
  "error": {
    ".tag": "property_field_too_large"
  }
}
```



≡

Example: does_not_fit_template

```
{  
    "error_summary": "does_not_fit_template/...",  
    "error": {  
        ".tag": "does_not_fit_template"  
    }  
}
```

UpdatePropertiesError (*union*)

This datatype comes from an imported namespace originally defined in the file_properties namespace. The value will be one of the following datatypes:

template_not_found *String*(min_length=1, pattern="(|ptid:).*)")
Template does not exist for the given identifier.

restricted_content *Void* You do not have permission to modify this template.

path *LookupError*

unsupported_folder *Void* This folder cannot be tagged. Tagging folders is not supported for team-owned templates.

property_field_too_large *Void* One or more of the supplied property field values is too large.

does_not_fit_template *Void* One or more of the supplied property fields does not conform to the template specifications.

property_group_lookup *LookUpPropertiesError*

/docs/archive

VERSION

1 ▾

DEPRECATED

DESCRIPTION

Marks the given Paper doc as archived.

This action can be performed or undone by anyone with edit permissions to the doc.

Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use /users/features/get_values. If the paper_as_files feature is enabled, then the user is running the new version of Paper.

This endpoint will be retired in September 2020. Refer to the [Paper Migration Guide](#) for more information.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

<https://api.dropboxapi.com/2/paper/docs/archive>

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/paper/docs/archive \  
    --header "Authorization: Bearer <get access token>" \  
    --header "Content-Type: application/json" \  
    --data "{\"doc_id\": \"uaSvRuxvnkFa12PTkBv5q\"}"
```

PARAMETERS

```
{  
    "doc_id": "uaSvRuxvnkFa12PTkBv5q"  
}
```

[RefPaperDoc](#)

doc_id *String* The Paper doc ID.

RETURNS

No return values.

ERRORS

Example: insufficient_permissions

```
{  
    "error_summary": "insufficient_permissions/...",  
    "error": {  
        ".tag": "insufficient_permissions"  
    }  
}
```



```
    "error_summary": "other/...",
    "error": {
        ".tag": "other"
    }
}
```

Example: doc_not_found

```
{
    "error_summary": "doc_not_found/...",
    "error": {
        ".tag": "doc_not_found"
    }
}
```

DocLookupError (*union*)

The value will be one of the following datatypes:

insufficient_permissions *Void* Your account does not have permissions to perform this action. This may be due to it only having access to Paper as files in the Dropbox filesystem. For more information, refer to the [Paper Migration Guide](#).

doc_not_found *Void* The required doc was not found.

/docs/create

VERSION

1

DEPRECATED

DESCRIPTION

Creates a new Paper doc with the provided content.

Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use /users/features/get_values. If the paper_as_files feature is enabled, then the user is running the new version of Paper.

This endpoint will be retired in September 2020. Refer to the [Paper Migration Guide](#) for more information.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

<https://api.dropboxapi.com/2/paper/docs/create>

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[Content-upload](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/paper/docs/create \
--header "Authorization: Bearer <get access token>" \
--header "Dropbox-API-Arg: {\"import_format\": \"markdown\"}" \
--header "Content-Type: application/octet-stream" \
--data-binary @local_file.txt
```

PARAMETERS

Example: Create new Paper doc (unfiled).

```
{
    "import_format": "markdown"
}
```

Example: Create new Paper doc inside Paper folder.

```
{
    "import_format": "html",
    "parent_folder_id": "e.gGYT6HSafpMej9bUv306GarglI76GeAM3yvfZvXH09u4mV"
}
```

PaperDocCreateArgs

import_format *ImportFormat* The format of provided data.

parent_folder_id *String?* The Paper folder ID where the Paper document should be created. The API user has to have write access to this folder or error is thrown. This field is optional.

RETURNS

```
{
    "doc_id": "uaSvRuxvnkFa12PTkBv5q",
    "revision": 456736745,
    "title": "Week one retention"
}
```

PaperDocCreateUpdateResult



title *String* The Paper doc title.

ERRORS Example: insufficient_permissions

```
{  
    "error_summary": "insufficient_permissions/...",  
    "error": {  
        ".tag": "insufficient_permissions"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: content_malformed

```
{  
    "error_summary": "content_malformed/...",  
    "error": {  
        ".tag": "content_malformed"  
    }  
}
```

Example: folder_not_found

```
{  
    "error_summary": "folder_not_found/...",  
    "error": {  
        ".tag": "folder_not_found"  
    }  
}
```

Example: doc_length_exceeded

```
{  
    "error_summary": "doc_length_exceeded/...",  
    "error": {  
        ".tag": "doc_length_exceeded"  
    }  
}
```

Example: image_size_exceeded

```
{  
    "error_summary": "image_size_exceeded/...",  
    "error": {  
        ".tag": "image_size_exceeded"  
    }  
}
```

PaperDocCreateError (*union*)

The value will be one of the following datatypes:

insufficient_permissions *Void* Your account does not have permissions to perform this action. This may be due to it only having access to Paper as files in the Dropbox filesystem. For more information, refer to the [Paper Migration Guide](#).

content_malformed *Void* The provided content was malformed and cannot be imported to Paper.

folder_not_found *Void* The specified Paper folder is cannot be found.

doc_length_exceeded *Void* The newly created Paper doc would be too large. Please split the content into multiple docs.

image_size_exceeded *Void* The imported document contains an image that is too large. The current limit is 1MB. This only applies to HTML with data URI.

/docs/download

VERSION

1 ▾

DEPRECATED

DESCRIPTION

Exports and downloads Paper doc either as HTML or markdown.

Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use /users/features/get_values. If the paper_as_files feature is enabled, then the user is running



This endpoint does not support apps with the app folder permission.

URL STRUCTURE

`https://api.dropboxapi.com/2/paper/docs/download`

AUTHENTICATION

User Authentication

ENDPOINT
FORMAT

Content-download

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/paper/docs/download \
  --header "Authorization: Bearer <get access token>" \
  --header "Dropbox-API-Arg: {\"doc_id\": \"uaSvRuxvnkFa12PTkBv5q\", \"export_format\": \"markdown\"}"
```

PARAMETERS

```
{  
  "doc_id": "uaSvRuxvnkFa12PTkBv5q",  
  "export_format": "markdown"  
}
```

PaperDocExport

doc_id *String* The Paper doc ID.

export_format *ExportFormat*

RETURNS

```
{  
  "owner": "james@example.com",  
  "title": "Week one retention",  
  "revision": 456736745,  
  "mime_type": "text/x-markdown"  
}
```

PaperDocExportResult

owner *String* The Paper doc owner's email address.

title *String* The Paper doc title.

revision *Int64* The Paper doc revision. Simply an ever increasing number.

mime_type *String* MIME type of the export. This corresponds to ExportFormat specified in the request.

ERRORS

Example: insufficient_permissions

```
{  
  "error_summary": "insufficient_permissions...",  
  "error": {  
    ".tag": "insufficient_permissions"  
  }  
}
```

Example: other

```
{  
  "error_summary": "other...",  
  "error": {  
    ".tag": "other"  
  }  
}
```

Example: doc_not_found

```
{  
  "error_summary": "doc_not_found...",  
  "error": {  
    ".tag": "doc_not_found"  
  }  
}
```

DocLookupError (*union*)

The value will be one of the following datatypes:

insufficient_permissions *Void* Your account does not have permissions to perform this action. This may be due to it only having access to Paper as files in the Dropbox filesystem. For more information, refer to the [Paper Migration Guide](#).

doc_not_found *Void* The required doc was not found.

/docs/folder_users/list

VERSION

1

DEPRECATED



and

Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use /users/features/get_values. If the paper_as_files feature is enabled, then the user is running the new version of Paper. Refer to the [Paper Migration Guide](#) for migration information.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

`https://api.dropboxapi.com/2/paper/docs/folder_users/list`

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/paper/docs/folder_users/list \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"doc_id\": \"uaSvRuxvnkFa12PTkBv5q\"},\"limit\": 100}"
```

PARAMETERS

```
{  
    "doc_id": "uaSvRuxvnkFa12PTkBv5q",  
    "limit": 100  
}
```

ListUsersOnFolderArgs

doc_id *String* The Paper doc ID.

limit *Int32(min=1, max=1000)* Size limit per batch. The maximum number of users that can be retrieved per batch is 1000. Higher value results in invalid arguments error. The default for this field is 1000.

RETURNS

```
{  
    "invitees": [  
        {  
            ".tag": "email",  
            "email": "jessica@example.com"  
        }  
    ],  
    "users": [  
        {  
            "account_id": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc",  
            "email": "bob@example.com",  
            "display_name": "Robert Smith",  
            "same_team": true,  
            "team_member_id": "dbmid:abcd1234"  
        }  
    ],  
    "cursor": {  
        "value": "zHZvTPBnXilGgm1AmDgVz10zf7qb0qznd5sAVQbbIVoteSnWLjUdLU7aR25hb",  
        "expiration": "2016-08-07T14:56:15Z"  
    },  
    "has_more": false  
}
```

ListUsersOnFolderResponse

invitees *List of (InviteeInfo)* List of email addresses that are invited on the Paper folder.

users *List of (UserInfo)* List of users that are invited on the Paper folder.

cursor *Cursor* Pass the cursor into [docs/folder_users/list/continue](#) to paginate through all users. The cursor preserves all properties as specified in the original call to [docs/folder_users/list](#).

has_more *Boolean* Will be set to True if a subsequent call with the provided cursor to [docs/folder_users/list/continue](#) returns immediately with some results. If set to False please allow some delay before making another call to [docs/folder_users/list/continue](#).

ERRORS Example: insufficient_permissions

```
{  
    "error_summary": "insufficient_permissions/...",  
    "error": {  
        ".tag": "insufficient_permissions"  
    }  
}
```

Example: other



```
        ".tag": "other"
    }
}
```

Example: doc not found

```
{  
    "error_summary": "doc_not_found...",  
    "error": {  
        ".tag": "doc_not_found"  
    }  
}
```

DocLookupError (*union*)

The value will be one of the following datatypes:

insufficient_permissions *Void* Your account does not have permissions to perform this action. This may be due to it only having access to Paper as files in the Dropbox filesystem. For more information, refer to the [Paper Migration Guide](#).

doc not found *Void* The required doc was not found.

/docs/folder users/list/continue

VERSION

1

DEPRECATED

DESCRIPTION Once a cursor has been retrieved from `docs/folder_users/list`, use this to paginate through all users on the Paper folder.

Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use `/users/features/get_values`. If the `paper_as_files` feature is enabled, then the user is running the new version of Paper.

Refer to the [Paper Migration Guide](#) for migration information.

Refer to the [Paper Migration Guide](#) for migration information.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/paper/docs/folder_users/list/continue

AUTHENTICATION

User Authentication

ENDPOINT FORMAT

EXAMPLE

Get access token for: ProfesorRobot

PARAMETERS

6

```
{ "doc_id": "uaSvRuxvnkFa12PTkBv5q",
  "cursor": "U60b6BxT43ySd5sAVQbbIVoteSnWLjUdLU7aR25hb3ySd5sAVQbbIVoteSnWLjUd"
}
```

ListUsersOnFolderContinueArgs

doc_id *String* The Paper doc ID.

cursor *String* The cursor obtained from `docs/folder_users/list` or `docs/folder_users/list/continue`. Allows for pagination.

RETURNS

```
{
  "invitees": [
    {
      ".tag": "email",
      "email": "jessica@example.com"
    }
  ],
  "users": [
    {
      "account_id": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc",
      "email": "bob@example.com",
      "display_name": "Robert Smith",
      "same_team": true,
      "team_member_id": "dbmid:abcd1234"
    }
  ],
  "cursor": {
    "value": "zHZvTPBnXilGgm1AmDgVyZl0zf7qb0qznd5sAVQbbIVoteSnWLjUdLU7aR25hb",
    "expiration": "2016-08-07T14:56:15Z"
  },
  "has_more": false
}
```



≡

invited on the Paper folder.

users *List of (UserInfo)* List of users that are invited on the Paper folder.

cursor *Cursor* Pass the cursor into [docs/folder_users/list/continue](#) to paginate through all users. The cursor preserves all properties as specified in the original call to [docs/folder_users/list](#).

has_more *Boolean* Will be set to True if a subsequent call with the provided cursor to [docs/folder_users/list/continue](#) returns immediately with some results. If set to False please allow some delay before making another call to [docs/folder_users/list/continue](#).

ERRORS

Example: insufficient_permissions

```
{  
    "error_summary": "insufficient_permissions/...",  
    "error": {  
        ".tag": "insufficient_permissions"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: doc_not_found

```
{  
    "error_summary": "doc_not_found/...",  
    "error": {  
        ".tag": "doc_not_found"  
    }  
}
```

ListUsersCursorError (*union*)

The value will be one of the following datatypes:

insufficient_permissions *Void* Your account does not have permissions to perform this action. This may be due to it only having access to Paper as files in the Dropbox filesystem. For more information, refer to the [Paper Migration Guide](#).

doc_not_found *Void* The required doc was not found.

cursor_error *PaperApiCursorError*

/docs/get_folder_info

VERSION

1

DEPRECATED

DESCRIPTION

Retrieves folder information for the given Paper doc. This includes:

- folder sharing policy; permissions for subfolders are set by the top-level folder.
- full 'filepath', i.e. the list of folders (both folderId and folderName) from the root folder to the folder directly containing the Paper doc.

If the Paper doc is not in any folder (aka unfiled) the response will be empty.

Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use [/users/features/get_values](#). If the paper_as_files feature is enabled, then the user is running the new version of Paper.

Refer to the [Paper Migration Guide](#) for migration information.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/paper/docs/get_folder_info

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/paper/docs/get_folder_info \  
--header "Authorization: Bearer <get access token>" \  
-
```



PARAMETERS

```
{  
    "doc_id": "uaSvRuxvnkFa12PTkBv5q"  
}
```

RefPaperDoc

doc_id *String* The Paper doc ID.

RETURNS

```
{  
    "folder_sharing_policy_type": {  
        ".tag": "team"  
    },  
    "folders": [  
        {  
            "id": "e.gYT6HSafpMej9bUv306oGm60vrHiCHgEFnzzioPGCvHf",  
            "name": "Design docs"  
        }  
    ]  
}
```

FoldersContainingPaperDoc

Metadata about Paper folders containing the specified Paper doc.

folder_sharing_policy_type *FolderSharingPolicyType?* The sharing policy of the folder containing the Paper doc. This field is optional.

folders *List of (Folder)?* The folder path. If present the first folder is the root folder. This field is optional.

ERRORS

Example: insufficient_permissions

```
{  
    "error_summary": "insufficient_permissions/...",  
    "error": {  
        ".tag": "insufficient_permissions"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: doc_not_found

```
{  
    "error_summary": "doc_not_found/...",  
    "error": {  
        ".tag": "doc_not_found"  
    }  
}
```

DocLookupError (*union*)

The value will be one of the following datatypes:

insufficient_permissions *Void* Your account does not have permissions to perform this action. This may be due to it only having access to Paper as files in the Dropbox filesystem. For more information, refer to the [Paper Migration Guide](#).

doc_not_found *Void* The required doc was not found.

/docs/list

VERSION

1 ▾

DEPRECATED

DESCRIPTION

Return the list of all Paper docs according to the argument specifications. To iterate over through the full pagination, pass the cursor to [docs/list/continue](#).

Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use /users/features/get_values. If the paper_as_files feature is enabled, then the user is running the new version of Paper.

Refer to the [Paper Migration Guide](#) for migration information.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

<https://api.dropboxapi.com/2/paper/docs/list>



FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/paper/docs/list \
  -header "Authorization: Bearer <get access token>" \
  -header "Content-Type: application/json" \
  --data "{\"filter_by\": \"docs_created\", \"sort_by\": \"modified\", \"sort_order\": \"descending\", \"limit\": 100}
```

PARAMETERS

```
{  
  "filter_by": "docs_created",  
  "sort_by": "modified",  
  "sort_order": "descending",  
  "limit": 100  
}
```

ListPaperDocsArgs

filter_by *ListPaperDocsFilterBy* Allows user to specify how the Paper docs should be filtered. The default for this union is docs_accessed.

sort_by *ListPaperDocsSortBy* Allows user to specify how the Paper docs should be sorted. The default for this union is accessed.

sort_order *ListPaperDocsSortOrder* Allows user to specify the sort order of the result. The default for this union is ascending.

limit *Int32(min=1, max=1000)* Size limit per batch. The maximum number of docs that can be retrieved per batch is 1000. Higher value results in invalid arguments error. The default for this field is 1000.

RETURNS

```
{  
  "doc_ids": [  
    "z01E7coc54sE8IuMdUoxz",  
    "mnlAmDgVyZl0zf7qb0qzn",  
    "dByYHZvTPBnXilGgyc5mm"  
,  
  "cursor": {  
    "value": "zHZvTPBnXilGgmc1AmDgVyZl0zf7qb0qznd5sAVQbbIVoteSnWLjUdLU7aR25hb",  
    "expiration": "2016-08-07T14:56:15Z"  
  },  
  "has_more": true  
}
```

ListPaperDocsResponse

doc_ids *List of (String)* The list of Paper doc IDs that can be used to access the given Paper docs or supplied to other API methods. The list is sorted in the order specified by the initial call to [docs/list](#).

cursor *Cursor* Pass the cursor into [docs/list/continue](#) to paginate through all files. The cursor preserves all properties as specified in the original call to [docs/list](#).

has_more *Boolean* Will be set to True if a subsequent call with the provided cursor to [docs/list/continue](#) returns immediately with some results. If set to False please allow some delay before making another call to [docs/list/continue](#).

ERRORS

No errors.

[/docs/list/continue](#)

VERSION

1 ▾

DEPRECATED

DESCRIPTION

Once a cursor has been retrieved from [docs/list](#), use this to paginate through all Paper doc. Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use [/users/features/get_values](#). If the paper_as_files feature is enabled, then the user is running the new version of Paper. Refer to the [Paper Migration Guide](#) for migration information.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

<https://api.dropboxapi.com/2/paper/docs/list/continue>

AUTHENTICATION

[User Authentication](#)ENDPOINT
FORMAT[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾



≡ \

--data "{\"cursor\": \"U60b6BxT43ySd5sAVQbbIvoteSnWLjUdLU7aR25hbt3ySd5sAVQbbIvoteSnWLjUd\"}"

PARAMETERS

```
{  
  "cursor": "U60b6BxT43ySd5sAVQbbIvoteSnWLjUdLU7aR25hbt3ySd5sAVQbbIvoteSnWLjUd"  
}
```

ListPaperDocsContinueArgs

cursor *String* The cursor obtained from [docs/list](#) or [docs/list/continue](#). Allows for pagination.

RETURNS

```
{  
  "doc_ids": [  
    "z01E7coc54sE8IuMdUoxz",  
    "mnlAmDgVyZ10zf7qb0qzn",  
    "dBvYHZvTPBnXilGgyc5mm"  
,  
  "cursor": {  
    "value": "zHZvTPBnXilGgm1AmDgVyZ10zf7qb0qzn5sAVQbbIvoteSnWLjUdLU7aR25hb",  
    "expiration": "2016-08-07T14:56:15Z"  
  },  
  "has_more": true  
}
```

ListPaperDocsResponse

doc_ids *List of (String)* The list of Paper doc IDs that can be used to access the given Paper docs or supplied to other API methods. The list is sorted in the order specified by the initial call to [docs/list](#).

cursor *Cursor* Pass the cursor into [docs/list/continue](#) to paginate through all files. The cursor preserves all properties as specified in the original call to [docs/list](#).

has_more *Boolean* Will be set to True if a subsequent call with the provided cursor to [docs/list/continue](#) returns immediately with some results. If set to False please allow some delay before making another call to [docs/list/continue](#).

ERRORS

ListDocsCursorError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

cursor_error *PaperApiCursorError*

/docs/permanently_delete

VERSION

1 ▾

DEPRECATED

DESCRIPTION

Permanently deletes the given Paper doc. This operation is final as the doc cannot be recovered.

This action can be performed only by the doc owner.

Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use [/users/features/get_values](#). If the paper_as_files feature is enabled, then the user is running the new version of Paper.

Refer to the [Paper Migration Guide](#) for migration information.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

```
https://api.dropboxapi.com/2/paper/docs/permanently_delete
```

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/paper/docs/permanently_delete \  
  --header "Authorization: Bearer <get access token>" \  
  --header "Content-Type: application/json" \  
  --data "{\"doc_id\": \"uaSvRuxvnkFa12PTkBv5q\"}"
```

PARAMETERS

```
{  
  "doc_id": "uaSvRuxvnkFa12PTkBv5q"  
}
```

RefPaperDoc

doc_id *String* The Paper doc ID.



ERRORS

Example: insufficient_permissions

```
{  
    "error_summary": "insufficient_permissions/...",  
    "error": {  
        ".tag": "insufficient_permissions"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: doc_not_found

```
{  
    "error_summary": "doc_not_found/...",  
    "error": {  
        ".tag": "doc_not_found"  
    }  
}
```

DocLookupError (*union*)

The value will be one of the following datatypes:

insufficient_permissions *Void* Your account does not have permissions to perform this action. This may be due to it only having access to Paper as files in the Dropbox filesystem. For more information, refer to the [Paper Migration Guide](#).

doc_not_found *Void* The required doc was not found.

/docs/sharing_policy/get

VERSION

1 ▾

DEPRECATED

DESCRIPTION

Gets the default sharing policy for the given Paper doc.

Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use /users/features/get_values. If the paper_as_files feature is enabled, then the user is running the new version of Paper.

Refer to the [Paper Migration Guide](#) for migration information.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

```
https://api.dropboxapi.com/2/paper/docs/sharing_policy/get
```

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/paper/docs/sharing_policy/get \  
-H "Authorization: Bearer <get access token>" \  
-H "Content-Type: application/json" \  
-d "{\"doc_id\": \"uaSvRuxvnkFa12PTkBv5q\"}"
```

PARAMETERS

```
{  
    "doc_id": "uaSvRuxvnkFa12PTkBv5q"  
}
```

RefPaperDoc

doc_id *String* The Paper doc ID.

RETURNS

```
{  
    "public_sharing_policy": {  
        ".tag": "people_with_link_can_edit"  
    },  
    "team_sharing_policy": {  
        ".tag": "people_with_link_can_edit"  
    }  
}
```

SharingPolicy



team_sharing_policy *SharingTeamPolicyType?* This value applies to the team members only. The value is null for all personal accounts. This field is optional.

ERRORS Example: insufficient_permissions

```
{  
    "error_summary": "insufficient_permissions/...",  
    "error": {  
        ".tag": "insufficient_permissions"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: doc_not_found

```
{  
    "error_summary": "doc_not_found/...",  
    "error": {  
        ".tag": "doc_not_found"  
    }  
}
```

DocLookupError (*union*)

The value will be one of the following datatypes:

insufficient_permissions *Void* Your account does not have permissions to perform this action. This may be due to it only having access to Paper as files in the Dropbox filesystem. For more information, refer to the [Paper Migration Guide](#).

doc_not_found *Void* The required doc was not found.

/docs/sharing_policy/set

VERSION

1 ▾

DEPRECATED

DESCRIPTION Sets the default sharing policy for the given Paper doc. The default 'team_sharing_policy' can be changed only by teams, omit this field for personal accounts.
The 'public_sharing_policy' policy can't be set to the value 'disabled' because this setting can be changed only via the team admin console.
Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use /users/features/get_values. If the paper_as_files feature is enabled, then the user is running the new version of Paper.
Refer to the [Paper Migration Guide](#) for migration information.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/paper/docs/sharing_policy/set

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/paper/docs/sharing_policy/set \  
--header "Authorization: Bearer <get access token> " \  
--header "Content-Type: application/json" \  
--data "{\"doc_id\": \"ua5vRuxvnkFa12PTkBv5q\", \"sharing_policy\": {\"public_sharing_policy\": \"people_with_link_can_edit\", \"team_sharing_policy\": \"people_with_link_can_edit\"}}
```

PARAMETERS

```
{  
    "doc_id": "ua5vRuxvnkFa12PTkBv5q",  
    "sharing_policy": {  
        "public_sharing_policy": "people_with_link_can_edit",  
        "team_sharing_policy": "people_with_link_can_edit"  
    }  
}
```

PaperDocSharingPolicy



≡ set

RETURNS No return values.

ERRORS Example: insufficient_permissions

```
{  
    "error_summary": "insufficient_permissions/...",  
    "error": {  
        ".tag": "insufficient_permissions"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: doc_not_found

```
{  
    "error_summary": "doc_not_found/...",  
    "error": {  
        ".tag": "doc_not_found"  
    }  
}
```

DocLookupError (*union*)

The value will be one of the following datatypes:

insufficient_permissions *Void* Your account does not have permissions to perform this action. This may be due to it only having access to Paper as files in the Dropbox filesystem. For more information, refer to the [Paper Migration Guide](#).

doc_not_found *Void* The required doc was not found.

/docs/update

VERSION

1 ▾

DEPRECATED

DESCRIPTION

Updates an existing Paper doc with the provided content.

Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use /users/features/get_values. If the paper_as_files feature is enabled, then the user is running the new version of Paper.

This endpoint will be retired in September 2020. Refer to the [Paper Migration Guide](#) for more information.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

```
https://api.dropboxapi.com/2/paper/docs/update
```

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[Content-upload](#)

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/paper/docs/update \  
    --header "Authorization: Bearer <get access token>" \  
    --header "Dropbox-API-Arg: {\"doc_id\": \"uaSvRuxvnkFa12PTkBv5q\", \"doc_update_policy\": \"overwrite_all\", \"re..."}" \  
    --header "Content-Type: application/octet-stream" \  
    --data-binary @local_file.txt
```

PARAMETERS

Example: Overwrite the doc content with provided content.

```
{  
    "doc_id": "uaSvRuxvnkFa12PTkBv5q",  
    "doc_update_policy": "overwrite_all",  
    "revision": 12345,  
    "import_format": "html"  
}
```

Example: Prepend the content into the doc (the doc title will remain unchanged).



```
        "revision": 55555,  
        "import_format": "plain_text"  
    }
```

PaperDocUpdateArgs

doc_id *String* The Paper doc ID.

doc_update_policy *PaperDocUpdatePolicy* The policy used for the current update call.

revision *Int64* The latest doc revision. This value must match the head revision or an error code will be returned. This is to prevent colliding writes.

import_format *ImportFormat* The format of provided data.

RETURNS

```
{  
    "doc_id": "uaSvRuxvnkFa12PTkBv5q",  
    "revision": 456736745,  
    "title": "Week one retention"  
}
```

PaperDocCreateUpdateResult

doc_id *String* Doc ID of the newly created doc.

revision *Int64* The Paper doc revision. Simply an ever increasing number.

title *String* The Paper doc title.

ERRORS

Example: insufficient_permissions

```
{  
    "error_summary": "insufficient_permissions/...",  
    "error": {  
        ".tag": "insufficient_permissions"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: doc_not_found

```
{  
    "error_summary": "doc_not_found/...",  
    "error": {  
        ".tag": "doc_not_found"  
    }  
}
```

Example: content_malformed

```
{  
    "error_summary": "content_malformed/...",  
    "error": {  
        ".tag": "content_malformed"  
    }  
}
```

Example: revision_mismatch

```
{  
    "error_summary": "revision_mismatch/...",  
    "error": {  
        ".tag": "revision_mismatch"  
    }  
}
```

Example: doc_length_exceeded

```
{  
    "error_summary": "doc_length_exceeded/...",  
    "error": {  
        ".tag": "doc_length_exceeded"  
    }  
}
```

Example: image_size_exceeded

```
{  
    "error_summary": "image_size_exceeded/...",  
    "error": {  
        ".tag": "image_size_exceeded"  
    }  
}
```



```
{  
    "error_summary": "doc_archived/...",  
    "error": {  
        ".tag": "doc_archived"  
    }  
}
```

Example: doc_deleted

```
{  
    "error_summary": "doc_deleted/...",  
    "error": {  
        ".tag": "doc_deleted"  
    }  
}
```

PaperDocUpdateError (*union*)

The value will be one of the following datatypes:

insufficient_permissions *Void* Your account does not have permissions to perform this action. This may be due to it only having access to Paper as files in the Dropbox filesystem. For more information, refer to the [Paper Migration Guide](#).

doc_not_found *Void* The required doc was not found.

content_malformed *Void* The provided content was malformed and cannot be imported to Paper.

revision_mismatch *Void* The provided revision does not match the document head.

doc_length_exceeded *Void* The newly created Paper doc would be too large, split the content into multiple docs.

image_size_exceeded *Void* The imported document contains an image that is too large. The current limit is 1MB. This only applies to HTML with data URI.

doc_archived *Void* This operation is not allowed on archived Paper docs.

doc_deleted *Void* This operation is not allowed on deleted Paper docs.

/docs/users/add

VERSION

1

DEPRECATED

DESCRIPTION Allows an owner or editor to add users to a Paper doc or change their permissions using their email address or Dropbox account ID.
The doc owner's permissions cannot be changed.
Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use /users/features/get_values. If the paper_as_files feature is enabled, then the user is running the new version of Paper.
Refer to the [Paper Migration Guide](#) for migration information.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

<https://api.dropboxapi.com/2/paper/docs/users/add>

AUTHENTICATION

[User Authentication](#)

RPC
FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/paper/docs/users/add \  
--header "Authorization: Bearer <get access token>" \  
--header "Content-Type: application/json" \  
--data "{\"doc_id\": \"uaSvRuxvnkFa12PTkBv5q\", \"members\": [{\"member\": {\".tag\": \"email\", \"email\": \"justin@example.com\"}, \"permission_level\": \"view_and_comment\"}], \"custom_message\": \"Welcome to Paper.\"}
```

PARAMETERS

```
{  
    "doc_id": "uaSvRuxvnkFa12PTkBv5q",  
    "members": [  
        {  
            "member": {  
                ".tag": "email",  
                "email": "justin@example.com"  
            },  
            "permission_level": "view_and_comment"  
        }  
    ],  
    "custom_message": "Welcome to Paper."  
}
```



AddPaperDocUser

doc_id *String* The Paper doc ID.

members *List of (AddMember, max_items=20)* User which should be added to the Paper doc. Specify only email address or Dropbox account ID.

custom_message *String?* A personal message that will be emailed to each successfully added member. This field is optional.

quiet *Boolean* Clients should set this to true if no email message shall be sent to added users. The default for this field is False.

RETURNS This route returns a list. This means the route can accept a homogenous list of the following types:

AddPaperDocUserMemberResult

Per-member result for [docs/users/add](#).

member *MemberSelector* One of specified input members.

result *AddPaperDocUserResult* The outcome of the action on this member.

ERRORS Example: insufficient_permissions

```
{  
    "error_summary": "insufficient_permissions/...",  
    "error": {  
        ".tag": "insufficient_permissions"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: doc_not_found

```
{  
    "error_summary": "doc_not_found/...",  
    "error": {  
        ".tag": "doc_not_found"  
    }  
}
```

DocLookupError (*union*)

The value will be one of the following datatypes:

insufficient_permissions *Void* Your account does not have permissions to perform this action. This may be due to it only having access to Paper as files in the Dropbox filesystem. For more information, refer to the [Paper Migration Guide](#).

doc_not_found *Void* The required doc was not found.

/docs/users/list

VERSION

1 ▾

DEPRECATED

DESCRIPTION

Lists all users who visited the Paper doc or users with explicit access. This call excludes users who have been removed. The list is sorted by the date of the visit or the share date.

The list will include both users, the explicitly shared ones as well as those who came in using the Paper url link.

Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use /users/features/get_values. If the paper_as_files feature is enabled, then the user is running the new version of Paper.

Refer to the [Paper Migration Guide](#) for migration information.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

<https://api.dropboxapi.com/2/paper/docs/users/list>

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE



```
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"doc_id\": \"uaSvRuxvnkFa12PTkBv5q\", \"limit\": 100, \"filter_by\": \"shared\"}"
```

PARAMETERS

```
{  
    "doc_id": "uaSvRuxvnkFa12PTkBv5q",  
    "limit": 100,  
    "filter_by": "shared"  
}
```

ListUsersOnPaperDocArgs

doc_id *String* The Paper doc ID.

limit *Int32(min=1, max=1000)* Size limit per batch. The maximum number of users that can be retrieved per batch is 1000. Higher value results in invalid arguments error. The default for this field is 1000.

filter_by *UserOnPaperDocFilter* Specify this attribute if you want to obtain users that have already accessed the Paper doc. The default for this union is shared.

RETURNS

```
{  
    "invitees": [  
        {  
            "invitee": {  
                ".tag": "email",  
                "email": "jessica@example.com"  
            },  
            "permission_level": {  
                ".tag": "edit"  
            }  
        }  
    ],  
    "users": [  
        {  
            "user": {  
                "account_id": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc",  
                "email": "bob@example.com",  
                "display_name": "Robert Smith",  
                "same_team": true,  
                "team_member_id": "dbmid:abcd1234"  
            },  
            "permission_level": {  
                ".tag": "view_and_comment"  
            }  
        }  
    ],  
    "doc_owner": {  
        "account_id": "dbid:AAH4f99T0taONIb-OurWxbNQ6ywGRopQngc",  
        "email": "bob@example.com",  
        "display_name": "Robert Smith",  
        "same_team": true,  
        "team_member_id": "dbmid:abcd1234"  
    },  
    "cursor": {  
        "value": "zHZvTPBnXilGgmlAmDgVyZ10zf7qb0qznd5sAVQbbIVoteSnWLjUdLU7aR25hb",  
        "expiration": "2016-08-07T14:56:15Z"  
    },  
    "has_more": false  
}
```

ListUsersOnPaperDocResponse

invitees *List of (InviteeInfoWithPermissionLevel)* List of email addresses with their respective permission levels that are invited on the Paper doc.

users *List of (UserInfoWithPermissionLevel)* List of users with their respective permission levels that are invited on the Paper folder.

doc_owner *UserInfo* The Paper doc owner. This field is populated on every single response.

cursor *Cursor* Pass the cursor into [docs/users/list/continue](#) to paginate through all users. The cursor preserves all properties as specified in the original call to [docs/users/list](#).

has_more *Boolean* Will be set to True if a subsequent call with the provided cursor to [docs/users/list/continue](#) returns immediately with some results. If set to False please allow some delay before making another call to [docs/users/list/continue](#).

ERRORS

Example: insufficient_permissions

```
{  
    "error_summary": "insufficient_permissions...",  
    "error": {  
        ".tag": "insufficient_permissions"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
}
```



≡

Example: doc_not_found

```
{  
    "error_summary": "doc_not_found/...",  
    "error": {  
        ".tag": "doc_not_found"  
    }  
}
```

DocLookupError (*union*)

The value will be one of the following datatypes:

insufficient_permissions *Void* Your account does not have permissions to perform this action. This may be due to it only having access to Paper as files in the Dropbox filesystem. For more information, refer to the [Paper Migration Guide](#).

doc_not_found *Void* The required doc was not found.

/docs/users/list/continue

VERSION

1

DEPRECATED

DESCRIPTION Once a cursor has been retrieved from [docs/users/list](#), use this to paginate through all users on the Paper doc.
Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use [/users/features/get_values](#). If the paper_as_files feature is enabled, then the user is running the new version of Paper.
Refer to the [Paper Migration Guide](#) for migration information.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

<https://api.dropboxapi.com/2/paper/docs/users/list/continue>

AUTHENTICATION

[User Authentication](#)

ENDPOINT FORMAT

[RPC](#)

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/paper/docs/users/list/continue \  
--header "Authorization: Bearer <get access token>" \  
--header "Content-Type: application/json" \  
--data "{\"doc_id\": \"uaSvRuxvnkFa12PTkBv5q\", \"cursor\": \"U60b6BxT43ySd5sAVQbbIvoteSnWLjUdLU7aR25hbt3ySd5sAVQbbIvoteSnWLjUd"
```

PARAMETERS

```
{  
    "doc_id": "uaSvRuxvnkFa12PTkBv5q",  
    "cursor": "U60b6BxT43ySd5sAVQbbIvoteSnWLjUdLU7aR25hbt3ySd5sAVQbbIvoteSnWLjUd"
```

ListUsersOnPaperDocContinueArgs

doc_id *String* The Paper doc ID.

cursor *String* The cursor obtained from [docs/users/list](#) or [docs/users/list/continue](#). Allows for pagination.

RETURNS

```
{  
    "invitees": [  
        {  
            "invitedee": {  
                ".tag": "email",  
                "email": "jessica@example.com"  
            },  
            "permission_level": {  
                ".tag": "edit"  
            }  
        }  
    ],  
    "users": [  
        {  
            "user": {  
                "account_id": "dbid:AAH4f99T0taONib-OurWxbNQ6ywGRopQngc",  
                "email": "bob@example.com",  
                "display_name": "Robert Smith",  
                "same_team": true,  
                "team_member_id": "dbmid:abcd1234"  
            },  
            "permission_level": {  
                ".tag": "view_and_comment"  
            }  
        }  
    ]  
}
```



```
        "display_name": "Robert Smith",
        "same_team": true,
        "team_member_id": "dbmid:abcd1234"
    },
    "cursor": {
        "value": "zHZvTPBnXilGgm1AmDgVyZ10zf7qb0qznd5sAVQbbIvoteSnWLjUdLU7aR25hb",
        "expiration": "2016-08-07T14:56:15Z"
    },
    "has_more": false
}
```

ListUsersOnPaperDocResponse

invitees *List of (InviteeInfoWithPermissionLevel)* List of email addresses with their respective permission levels that are invited on the Paper doc.

users *List of (UserInfoWithPermissionLevel)* List of users with their respective permission levels that are invited on the Paper folder.

doc_owner *Userinfo* The Paper doc owner. This field is populated on every single response.

cursor *Cursor* Pass the cursor into [docs/users/list/continue](#) to paginate through all users. The cursor preserves all properties as specified in the original call to [docs/users/list](#).

has_more *Boolean* Will be set to True if a subsequent call with the provided cursor to [docs/users/list/continue](#) returns immediately with some results. If set to False please allow some delay before making another call to [docs/users/list/continue](#).

ERRORS Example: insufficient_permissions

```
{
    "error_summary": "insufficient_permissions...",
    "error": {
        ".tag": "insufficient_permissions"
    }
}
```

Example: other

```
{
    "error_summary": "other...",
    "error": {
        ".tag": "other"
    }
}
```

Example: doc_not_found

```
{
    "error_summary": "doc_not_found...",
    "error": {
        ".tag": "doc_not_found"
    }
}
```

ListUsersCursorError (*union*)

The value will be one of the following datatypes:

insufficient_permissions *Void* Your account does not have permissions to perform this action. This may be due to it only having access to Paper as files in the Dropbox filesystem. For more information, refer to the [Paper Migration Guide](#).

doc_not_found *Void* The required doc was not found.

cursor_error *PaperApiCursorError*

/docs/users/remove

VERSION

1

DEPRECATED

DESCRIPTION

Allows an owner or editor to remove users from a Paper doc using their email address or Dropbox account ID.

The doc owner cannot be removed.

Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use /users/features/get_values. If the paper_as_files feature is enabled, then the user is running the new version of Paper.

Refer to the [Paper Migration Guide](#) for migration information.

This endpoint does not support apps with the app folder permission.



AUTHENTICATION

User Authentication

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/paper/docs/users/remove \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"doc_id\": \"uaSvRuxvnkFa12PTkBv5q\", \"member\": {\".tag\": \"email\", \"email\": \"justin@example.com\"}}
```

PARAMETERS

```
{
  "doc_id": "uaSvRuxvnkFa12PTkBv5q",
  "member": {
    ".tag": "email",
    "email": "justin@example.com"
  }
}
```

RemovePaperDocUser

doc_id *String* The Paper doc ID.**member** *MemberSelector* User which should be removed from the Paper doc. Specify only email address or Dropbox account ID.

RETURNS

No return values.

ERRORS

Example: insufficient_permissions

```
{
  "error_summary": "insufficient_permissions/...",
  "error": {
    ".tag": "insufficient_permissions"
  }
}
```

Example: other

```
{
  "error_summary": "other/...",
  "error": {
    ".tag": "other"
  }
}
```

Example: doc_not_found

```
{
  "error_summary": "doc_not_found/...",
  "error": {
    ".tag": "doc_not_found"
  }
}
```

DocLookupError (*union*)

The value will be one of the following datatypes:

insufficient_permissions *Void* Your account does not have permissions to perform this action. This may be due to it only having access to Paper as files in the Dropbox filesystem. For more information, refer to the [Paper Migration Guide](#).**doc_not_found** *Void* The required doc was not found.[/folders/create](#)

VERSION

1 ▾

DEPRECATED

DESCRIPTION

Create a new Paper folder with the provided info.

Note that this endpoint will continue to work for content created by users on the older version of Paper. To check which version of Paper a user is on, use [/users/features/get_values](#). If the paper_as_files feature is enabled, then the user is running the new version of Paper.
Refer to the [Paper Migration Guide](#) for migration information.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

<https://api.dropboxapi.com/2/paper/folders/create>

AUTHENTICATION

User Authentication



```
curl -X POST https://api.dropboxapi.com/2/paper/folders/create \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"name\": \"my new folder\"}"
```

PARAMETERS Example: Create a top-level private folder.

```
{  
    "name": "my new folder"  
}
```

Example: Create a top-level team folder.

```
{  
    "name": "my new folder",  
    "is_team_folder": true  
}
```

Example: Create a new Paper folder inside an existing folder. The new folder will be a private or team folder depending on its parent.

```
{  
    "name": "my new folder",  
    "parent_folder_id": "e.gYT6HSafpMej9bUv306GarglI7GGAM3yvfZvXH09u4mV"  
}
```

PaperFolderCreateArg

name *String* The name of the new Paper folder.

parent_folder_id *String?* The encrypted Paper folder Id where the new Paper folder should be created. The API user has to have write access to this folder or error is thrown. If not supplied, the new folder will be created at top level. This field is optional.

is_team_folder *Boolean?* Whether the folder to be created should be a team folder. This value will be ignored if parent_folder_id is supplied, as the new folder will inherit the type (private or team folder) from its parent. We will by default create a top-level private folder if both parent_folder_id and is_team_folder are not supplied. This field is optional.

RETURNS

```
{  
    "folder_id": "abcd"  
}
```

PaperFolderCreateResult

folder_id *String* Folder ID of the newly created folder.

ERRORS

Example: insufficient_permissions

```
{  
    "error_summary": "insufficient_permissions/...",  
    "error": {  
        ".tag": "insufficient_permissions"  
    }  
}
```

Example: other

```
{  
    "error_summary": "other/...",  
    "error": {  
        ".tag": "other"  
    }  
}
```

Example: folder_not_found

```
{  
    "error_summary": "folder_not_found/...",  
    "error": {  
        ".tag": "folder_not_found"  
    }  
}
```

Example: invalid_folder_id

```
{  
    "error_summary": "invalid_folder_id/...",  
    "error": {  
        ".tag": "invalid_folder_id"  
    }  
}
```

PaperFolderCreateError (*union*)

The value will be one of the following datatypes:



information, refer to the [Paper Migration Guide](#).

folder_not_found *Void* The specified parent Paper folder cannot be found.

invalid_folder_id *Void* The folder id cannot be decrypted to valid folder id.

/change_file_member_access

VERSION

1

DEPRECATED BY /update_file_member

DESCRIPTION

Identical to update_file_member but with less information returned.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/change_file_member_access

AUTHENTICATION

User Authentication

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/sharing/change_file_member_access \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"file\": \"id:3kmLmQFnf1AAAAAAAAsAw\", \"member\": {\"tag\": \"email\", \"email\": \"justin@example.com\"}, \"access_level\": \"viewer\"}
```

PARAMETERS

```
{ "file": "id:3kmLmQFnf1AAAAAAAAsAw",
  "member": {
    ".tag": "email",
    "email": "justin@example.com"
  },
  "access_level": "viewer"
}
```

ChangeFileMemberAccessArgs

Arguments for [change_file_member_access](#).

file *String(min_length=1, pattern="((/id:).*)|nspath:[0-9]+::*)|ns:[0-9]+/(.*?)"* File for which we are changing a member's access.

member *MemberSelector* The member whose access we are changing.

access_level *AccessLevel* The new access level for the member.

RETURNS

```
{ "member": {
    ".tag": "email",
    "email": "justin@example.com"
  },
  "result": {
    ".tag": "success"
  }
}
```

FileMemberActionResult

Per-member result for [add_file_member](#) or [change_file_member_access](#).

member *MemberSelector* One of specified input members.

result *FileMemberActionIndividualResult* The outcome of the action on this member.

ERRORS

Example: invalid_member

```
{ "error_summary": "invalid_member/...",
  "error": {
    ".tag": "invalid_member"
  }
}
```

Example: no_permission

```
{ "error_summary": "no_permission/...",
  "error": {
    ".tag": "no_permission"
  }
}
```



```
"error_summary": "other/...",
"error": {
  ".tag": "other"
}
```

FileMemberActionError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

invalid_member *Void* Specified member was not found.

no_permission *Void* User does not have permission to perform this action on this member.

access_error *SharingFileAccessError* Specified file was invalid or user does not have access.

no_explicit_access *MemberAccessLevelResult* The action cannot be completed because the target member does not have explicit access to the file. The return value is the access that the member has to the file from a parent folder.

/create_shared_link

VERSION

1

DEPRECATED BY [/create_shared_link_with_settings](#)

DESCRIPTION

Create a shared link.

If a shared link already exists for the given path, that link is returned.

Note that in the returned PathLinkMetadata, the PathLinkMetadata.url field is the shortened URL if CreateSharedLinkArg.short_url argument is set to true.

Previously, it was technically possible to break a shared link by moving or renaming the corresponding file or folder. In the future, this will no longer be the case, so your app shouldn't rely on this behavior. Instead, if your app needs to revoke a shared link, use [revoke_shared_link](#).

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/create_shared_link

AUTHENTICATION

[User Authentication](#), [Dropbox-API-Select-Admin](#) (Team Admin)

ENDPOINT FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot

```
curl -X POST https://api.dropboxapi.com/2/sharing/create_shared_link \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"path\": \"/Homework/Math/Prime_Numbers.txt\", \"short_url\": false}"
```

PARAMETERS

```
{
  "path": "/Homework/Math/Prime_Numbers.txt",
  "short_url": false
}
```

CreateSharedLinkArg

path *String* The path to share.

short_url *Boolean* Whether to return a shortened URL. The default for this field is False.

pending_upload *PendingUploadMode?* If it's okay to share a path that does not yet exist, set this to either PendingUploadMode.file or PendingUploadMode.folder to indicate whether to assume it's a file or folder. This field is optional.

RETURNS

```
{
  "url": "https://www.dropbox.com/s/2sn712vy1ovegw8/Prime_Numbers.txt?dl=0",
  "visibility": {
    ".tag": "public"
  },
  "path": "/Homework/Math/Prime_Numbers.txt"
}
```

PathLinkMetadata

Metadata for a path-based shared link.

url *String* URL of the shared link.

visibility *Visibility* Who can access the link.

path *String* Path in user's Dropbox.

expires *Timestamp(format="%Y-%m-%dT%H:%M:%SZ")?*

Expiration time, if set. By default the link won't expire. This field is optional.

**path** *LookupError*

/get_shared_links

VERSION

1 ▾

DEPRECATED BY [/list_shared_links](#)

DESCRIPTION Returns a list of LinkMetadata objects for this user, including collection links.
If no path is given, returns a list of all shared links for the current user, including collection links, up to a maximum of 1000 links.
If a non-empty path is given, returns a list of all shared links that allow access to the given path. Collection links are never returned in this case.
Note that the url field in the response is never the shortened URL.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/get_shared_links

AUTHENTICATION

[User Authentication](#)

ENDPOINT
FORMAT

RPC

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/get_shared_links \
--header "Authorization: Bearer <get access token>" \
--header "Content-Type: application/json" \
--data "{\"path\": \"\"}"
```

PARAMETERS

Example: Get all links, including collection links.

```
{  
    "path": ""  
}
```

Example: Get links giving access to /Homework/Math.

```
{  
    "path": "/Homework/Math"  
}
```

GetSharedLinksArg

path *String?* See [get_shared_links](#) description. This field is optional.

RETURNS

```
{  
    "links": [  
        {  
            ".tag": "path",  
            "url": "https://www.dropbox.com/s/2sn712vy1ovegw8/Prime_Numbers.txt?dl=0",  
            "visibility": {  
                ".tag": "public"  
            },  
            "path": "/Homework/Math/Prime_Numbers.txt"  
        }  
    ]  
}
```

GetSharedLinksResult

links *List of (LinkMetadata)* Shared links applicable to the path argument.

ERRORS

GetSharedLinksError (*open union*)

The value will be one of the following datatypes. New values may be introduced as our API evolves.

path *String?* This field is optional.

/remove_file_member

VERSION

1 ▾

DEPRECATED BY [/remove_file_member_2](#)

DESCRIPTION

Identical to remove_file_member_2 but with less information returned.

This endpoint does not support apps with the app folder permission.

URL STRUCTURE

https://api.dropboxapi.com/2/sharing/remove_file_member



FORMAT

EXAMPLE

Get access token for: ProfesorRobot ▾

```
curl -X POST https://api.dropboxapi.com/2/sharing/remove_file_member \
  -header "Authorization: Bearer <get access token>" \
  -header "Content-Type: application/json" \
  -data "{\"file\": \"id:3kmLmQFnf1AAAAAAAAAAw\", \"member\": {\".tag\": \"email\", \"email\": \"justin@example.com\"}}
```

PARAMETERS

```
{  
  \"file\": \"id:3kmLmQFnf1AAAAAAAAAAw\",  
  \"member\": {  
    \".tag\": \"email\",  
    \"email\": \"justin@example.com\"  
  }  
}
```

RemoveFileMemberArg

Arguments for `remove_file_member_2`.

file *String(min_length=1, pattern="((/[id:].*)|nspath:[0-9]+(:.*))|ns:[0-9]+(:.*)?")* File from which to remove members.

member *MemberSelector* Member to remove from this file. Note that even if an email is specified, it may result in the removal of a user (not an invitee) if the user's main account corresponds to that email address.

RETURNS

```
{  
  \".tag\": \"success\"  
}
```

FileMemberActionIndividualResult (*union*)

The value will be one of the following datatypes:

success *AccessLevel?* Member was successfully removed from this file. If AccessLevel is given, the member still has access via a parent shared folder. This field is optional.

member_error *FileMemberActionError* User was not able to perform this action.

ERRORS

RemoveFileMemberError (*open union*)

Errors for `remove_file_member_2`. The value will be one of the following datatypes. New values may be introduced as our API evolves.

user_error *SharingUserError*

access_error *Sharing FileAccess Error*

no_explicit_access *MemberAccessLevelResult* This member does not have explicit access to the file and therefore cannot be removed. The return value is the access that a user might have to the file from a parent folder.

- Dropbox

Install
Mobile
Pricing
Business
Enterprise
Features

- About us

Dropbox Blog
About
Branding
News
Jobs

- Support

Help Center
Contact us
Copyright
Cookies
Privacy & Terms
Sitemap

- Community

Referrals
Forum



English (United States)