

Clouldinary

[Uploading with a direct call to the API](#)

Documentation:

Link: https://cloudinary.com/documentation/upload_images#uploading_with_a_direct_call_to_the_api

Uploading with a direct call to the API

Cloudinary's client libraries (SDKs) wrap the Upload API and greatly simplify using the API methods (see the [Upload API reference](#) documentation for more information on all the Upload API's methods). Nevertheless, if you wish to upload files with a direct call to the API from within your own custom code you can send an HTTPS POST request to the following Cloudinary URL:

```
https://api.cloudinary.com/v1_1/<cloud_name>/<resource_type>/upload
```

Where:

- `cloud_name` is the name of your Cloudinary account.
- `resource_type` is the type of file to upload. Valid values: `image`, `raw`, `video` and `auto` to automatically detect the file type.

For example, to upload an image file to the Cloudinary 'demo' account, send an HTTPS POST request to the following URL:

```
https://api.cloudinary.com/v1_1/demo/image/upload
```

Send Ajax POST request:

The contents of the POST request you send to Cloudinary depends on whether or not you are making an authenticated request (see the documentation on [Unsigned uploads](#) for more information on unauthenticated requests).

NOTE: More easy to use the unauthenticated method:

Required parameters for unauthenticated requests:

- `file` - The file to upload. Can be the actual data (byte array buffer), the Data URI (Base64 encoded), a remote FTP, HTTP or HTTPS URL of an existing file, or a private storage bucket (S3 or Google Storage) URL of a [whitelisted bucket](#).
- `upload_preset` - The name of an unsigned [upload preset](#) that you defined for [unsigned uploading](#) to your Cloudinary account.

Optional parameters:

Set the “upload_preset”

Unsigned upload

Unsigned upload is an option for performing upload directly from a browser or mobile application with no authentication signature, and without going through your servers at all. However, for security reasons, not all upload parameters can be specified directly when performing unsigned upload calls.

Unsigned upload options are controlled by an [upload preset](#), so in order to use this feature you first need to enable unsigned uploading for your Cloudinary account from the [Upload Settings](#) page. Enabling unsigned uploading also creates an upload preset with a unique name, which explicitly allows uploading of images without authentication. The preset is also used to define which upload options will be applied to images that are uploaded unsigned with that preset specified. You can edit the preset at any point in time (or create additional upload presets), to define the parameters that will be used for all images that are uploaded unsigned from user browsers or mobile applications.

For more information on upload presets, see the [upload preset](#) documentation and the [Centralized control for image upload](#) blog post.

Set the “upload_preset”

Go to “settings->upload”

Settings

Account **Upload** Security Users

Free Plan Upgrade plan 10 GB 1%

Automatic backup: Disabled

Enabling automatic backup means that every uploaded file is also copied to a secondary write-protected location, while keeping multiple revisions per file. Storage usage is increased when backup is enabled.

Backup S3 bucket: Automatic backup to an S3 bucket owned by you is available for our paid plans only. [Upgrade plan](#).

Auto-create folders: Enabled

When enabled, media library folders are automatically created to match the folders of uploaded images.

Invalidate versioned URLs: Disabled

Enable this if you use URLs with the version component, e.g.
<https://res.cloudinary.com/dcwibf9o5/image/upload/v1532614294/sample.jpg>

Auto upload mapping:

Folder: →

URL prefix:

Support Activities

- [Bulk delete](#)

Support Request

- [Allow uploading d](#)

Setup CNAME for my
Advanced plan or higher.

Manage user roles
Advanced plan or higher.

Enable Search API
Advanced Extra plan or hi

Usage Limits

- Upload API - Hourly r
- Admin API - Hourly r

Set the “upload_preset”

Find the upload presets and click on “Enable unsigned upload”

Upload presets:

[Enable unsigned uploading](#)

Simplify your image uploading procedure by enabling users to upload images and other assets into your Cloudinary account without pre-signing the upload request. For security reasons, unsigned uploads require using an upload preset.

[Add upload preset](#)


Upload presets allow you to define the default behavior for your uploads, instead of receiving these as parameters during the upload request itself. Parameters can include tags, incoming or on-demand transformations, notification URL, and more. Upload presets have precedence over client-side upload parameters.

Set the “upload_preset”

Copy the name

Upload presets:


Unsigned uploading enabled

Name	Mode	Settings	
vt0iqgff	Unsigned	Use filename: false	Edit Duplicate 

[Add upload preset](#)

Upload presets allow you to define the default behavior for your uploads, instead of receiving these as parameters during the upload request itself. Parameters can include tags, incoming or on-demand transformations, notification URL, and more. Upload presets have precedence over client-side upload parameters.

Cloud name:

 Dashboard Media Library Transformations Reports Add-ons Welcome 4/4

Dashboard

Free Plan Upgrade plan

Account Details

Download: YML, PY

Cloud name:	dcwibf9o5	
API Key:	263686384399685	Copy to clipboard
API Secret:	*****	Copy to clipboard Reveal
Environment variable:	CLOUDINARY_URL=cloudinary://*****:*****@dcwibf9o5	
Base delivery URL:	http://res.cloudinary.com/dcwibf9o5 ▼	
Secure delivery URL:	https://res.cloudinary.com/dcwibf9o5 ▼	
API Base URL:	https://api.cloudinary.com/v1_1/dcwibf9o5 ▼	

Your code:

```
function doUploadImg(elForm, onSuccess) {
  var UPLOAD_URL = `https://api.cloudinary.com/v1_1/${CLOUD_NAME}/image/upload`
  var UPLOAD_PRESET = "YOUR 'upload_preset'"
  var formData = new FormData();
  formData.append('file', elForm[0].files[0])
  formData.append('upload_preset', UPLOAD_PRESET);

  fetch(UPLOAD_URL, {
    method: 'POST',
    body: formData
  })
  .then(function (response) {
    return response.json()
  })
  .then(onSuccess)
  .catch(function (error) {
    console.error(error)
  })
}
```