Cloudinary academy

Introduction to Cloudinary's Android SDK

Jen Brissman
Technical Curriculum Engineer

Cloudinary

Topics

Introduction

- Mobile Development
- Android Course App
- Environment Setup

Upload

- Uploading an Asset
- Cloudinary Upload Widget
- Pre-Processing

Asset Delivery

- Transformations
- Optimizations

Video

- Cloudinary Native Video Player
- Creating a Video Feed

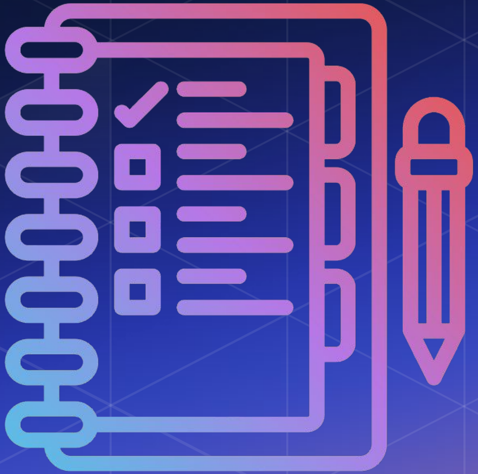
Download - Image Loaders

- Glide
- Picasso
- Fresco

Integration

- Glide Integration

Further Support



Mobile Statistics

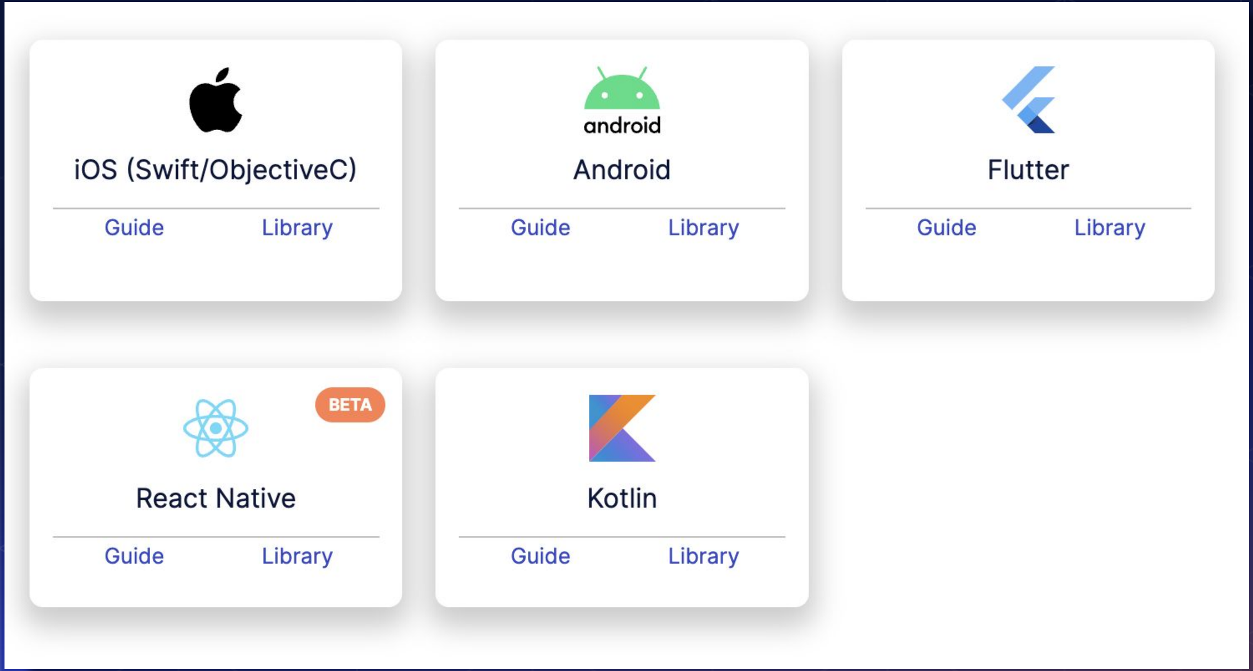
There are currently **6.3 billion** Smartphone users in the world. The mobile traffic market share is **58.22%** as of January 2024 beating out desktop at 38.75% and tablet at 2.03%. Yearly worldwide app downloads are at a staggering **257 billion**.

The demand for mobile development is at an all time high.

- Mobile apps are expected to generate over $935 billion in revenue in 2024.
- The Apple App Store has 1.96 million apps available for download.
- There are 2.87 million apps available for download on the Google Play Store.
- 21% of Millennials open an app 50+ times per day.
- 49% of people open an app 11+ times each day.
- 70% of all US digital media time comes from mobile apps.
- The average smartphone owner uses 10 apps per day and 30 apps each month.

Cloudinary Mobile SDKs

Cloudinary offers Mobile SDKs for both **iOS and Android** platforms, providing comprehensive image and video management capabilities for mobile applications.



iOS (Swift/ObjectiveC)
Guide Library

Android
Guide Library

Flutter
Guide Library

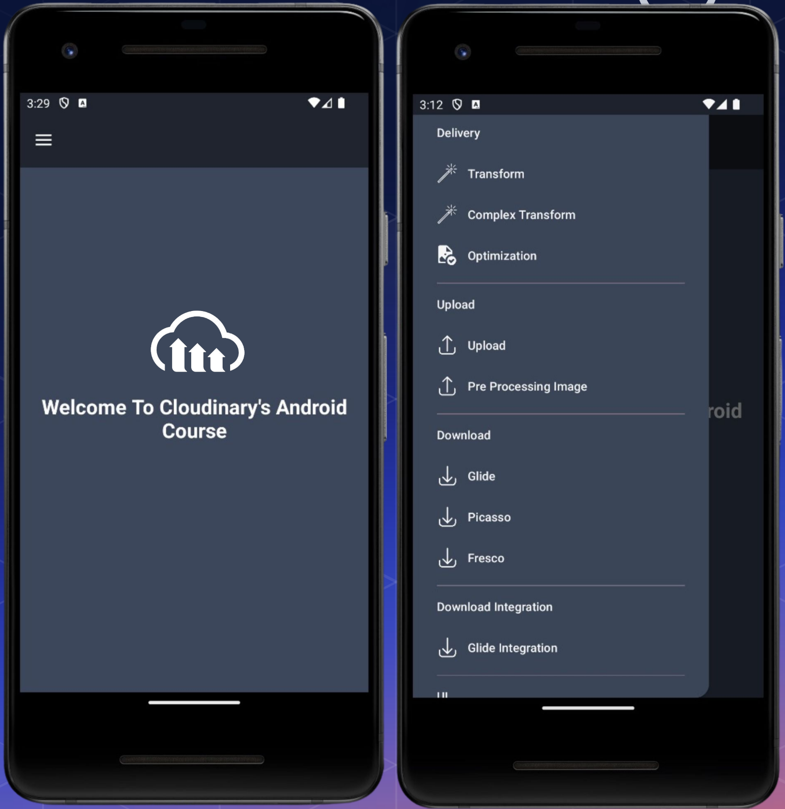
React Native (BETA)
Guide Library

Kotlin
Guide Library

Cloudinary

Cloudinary Android Course App

We have built an application specifically for this course which allows users to programmatically manipulate media assets.



Welcome To Cloudinary's Android Course

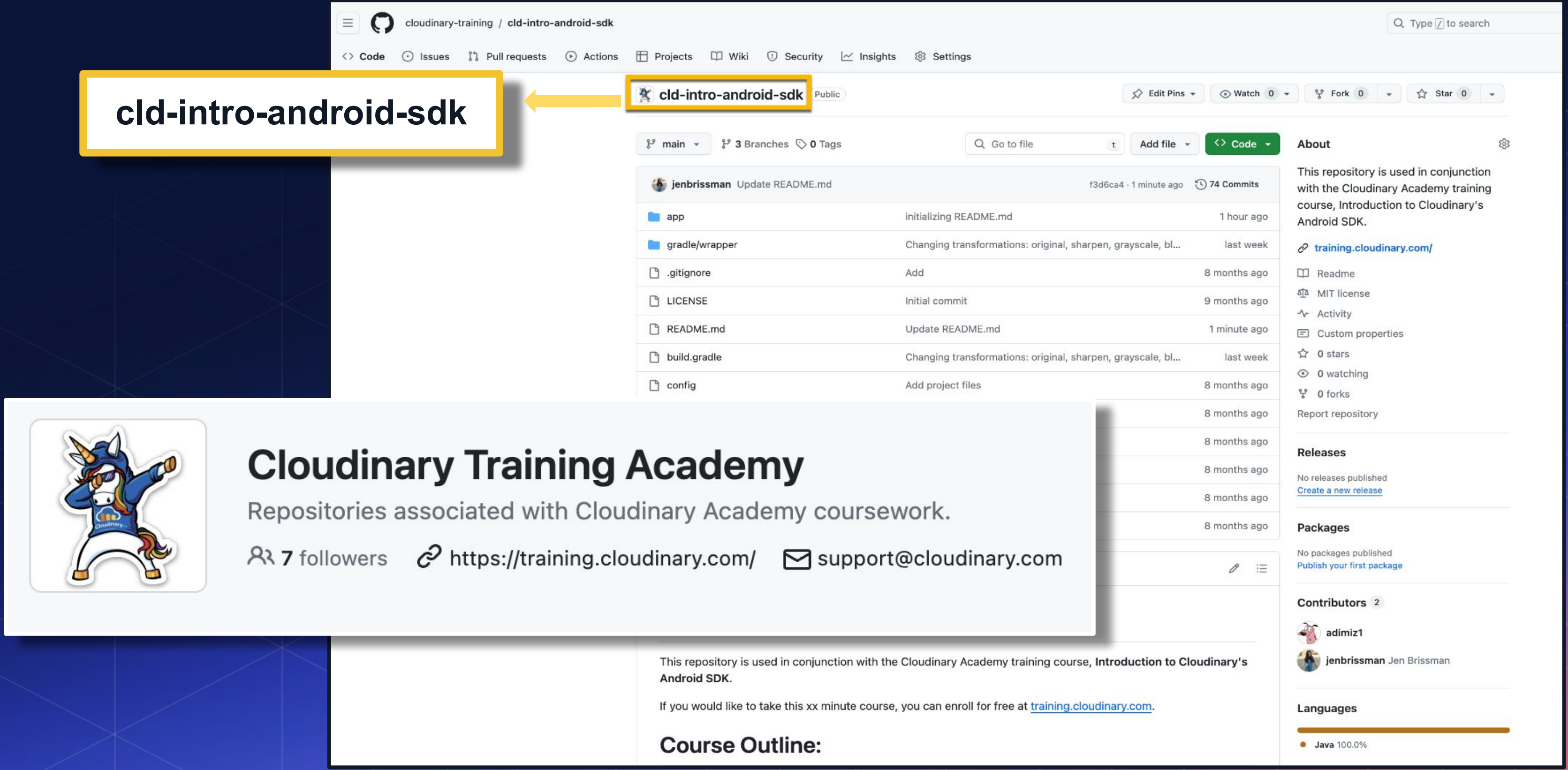
Cloudinary

Setting Up Your Environment

GitHub Repository

You can access all of the code for this course from the **cloudinary-training** GitHub.

cld-intro-android-sdk



cloudinary-training / **cld-intro-android-sdk**

cld-intro-android-sdk Public

jenbrissman Update README.md f3d6ca4 · 1 minute ago 74 Commits

app	initializing README.md	1 hour ago
gradle/wrapper	Changing transformations: original, sharpen, grayscale, bl...	last week
.gitignore	Add	8 months ago
LICENSE	Initial commit	9 months ago
README.md	Update README.md	1 minute ago
build.gradle	Changing transformations: original, sharpen, grayscale, bl...	last week
config	Add project files	8 months ago
		8 months ago
		8 months ago
		8 months ago
		8 months ago
		8 months ago

This repository is used in conjunction with the Cloudinary Academy training course, **Introduction to Cloudinary's Android SDK**.

If you would like to take this xx minute course, you can enroll for free at training.cloudinary.com.

Course Outline:

About

This repository is used in conjunction with the Cloudinary Academy training course, Introduction to Cloudinary's Android SDK.

training.cloudinary.com/

- Readme
- MIT license
- Activity
- Custom properties
- 0 stars
- 0 watching
- 0 forks

Report repository

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 2

- adimiz1
- jenbrissman Jen Brissman

Languages

Java 100.0%

Cloudinary Training Academy

Repositories associated with Cloudinary Academy coursework.

7 followers https://training.cloudinary.com/ support@cloudinary.com

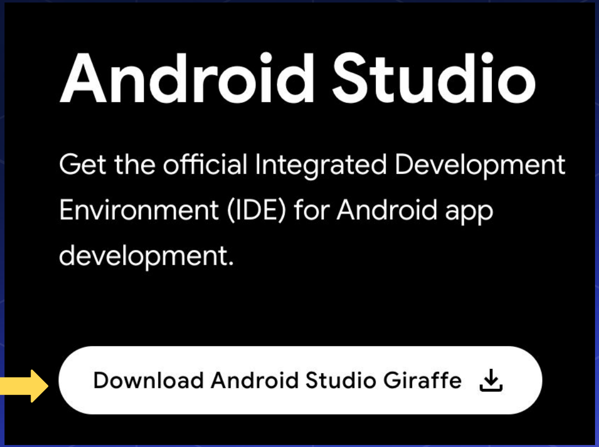
Install Java

Java Download
www.oracle.com/java/technologies/javase/javase-jdk8-downloads

macOS x64	**205.26 MB**	jdk-8u281-macosx-x64.dmg
Windows x64	**166.97 MB**	jdk-8u391-windows-x64.exe

```
$ java -version
java version "1.8.0_281"
Java(TM) SE Runtime Environment (build 1.8.0_281-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.281-b09, mixed mode)
```

Install Android Studio



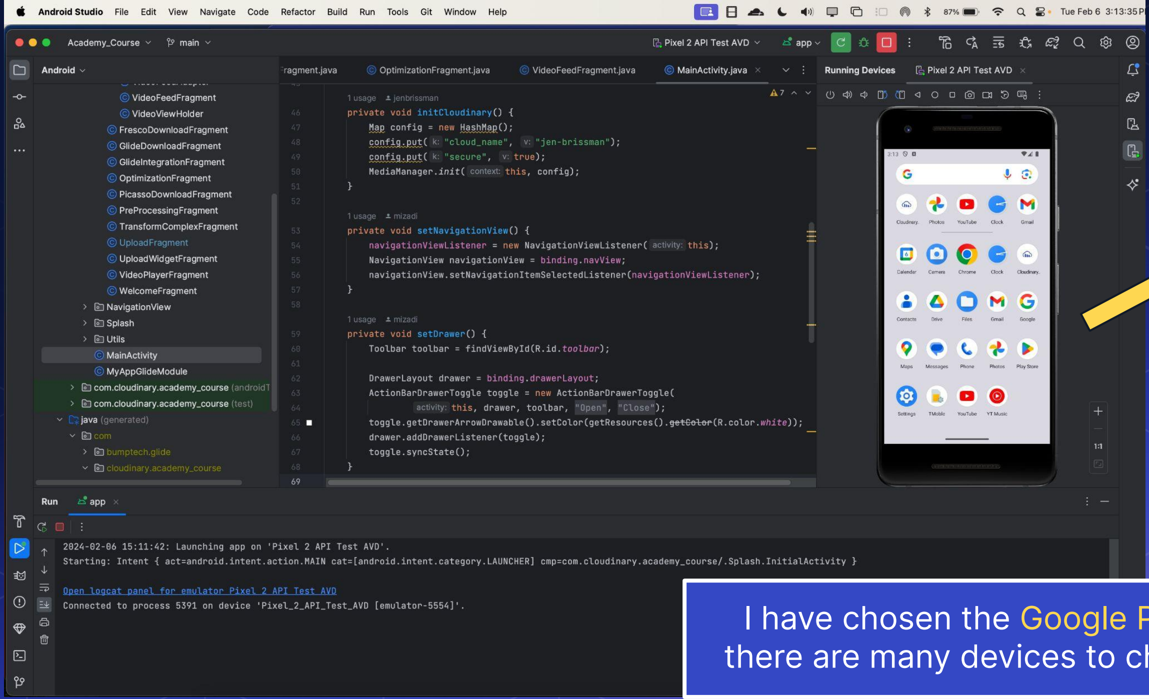
Android Studio

Get the official Integrated Development Environment (IDE) for Android app development.

https://developer.android.com/studio

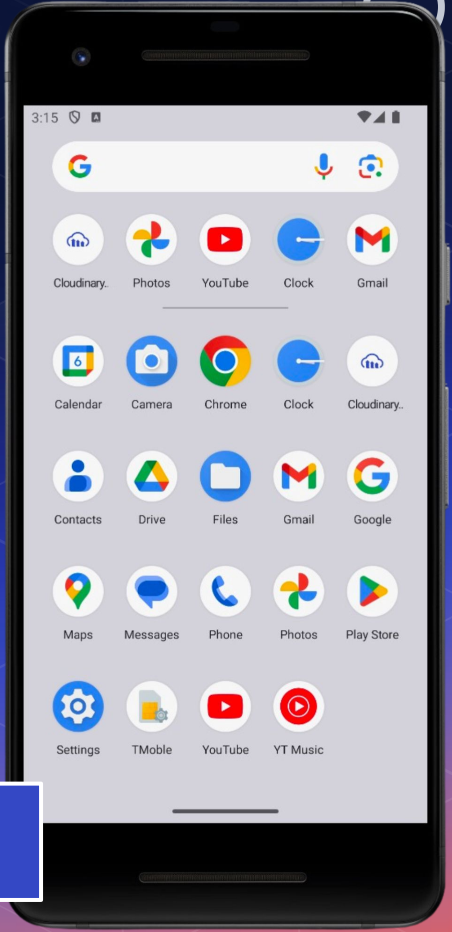
Android Emulator

The Android Emulator simulates Android devices within the IDE on your computer so that you can test your application on a variety of devices and Android API levels without needing to have each physical device.



```
    1 usage  jenbrissman
    private void initCloudinary() {
        Map config = new HashMap();
        config.put( k: "cloud_name",  v: "jen-brissman");
        config.put( k: "secure",  v: true);
        MediaManager.init( context: this, config);
    }

    1 usage  mizadi
    private void setNavigationView() {
        navigationViewListener = new NavigationViewListener( activity: this);
        NavigationView navigationView = binding.navView;
        navigationView.setNavigationItemSelectedListener(navigationViewListener);
    }

    1 usage  mizadi
    private void setDrawer() {
        Toolbar toolbar = findViewById(R.id.toolbar);

        DrawerLayout drawer = binding.drawerLayout;
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
                activity: this, drawer, toolbar, "Open", "Close");
        toggle.getDrawerArrowDrawable().setColor(getResources().~~getColor~~(R.color.white));
        drawer.addDrawerListener(toggle);
        toggle.syncState();
    }
```

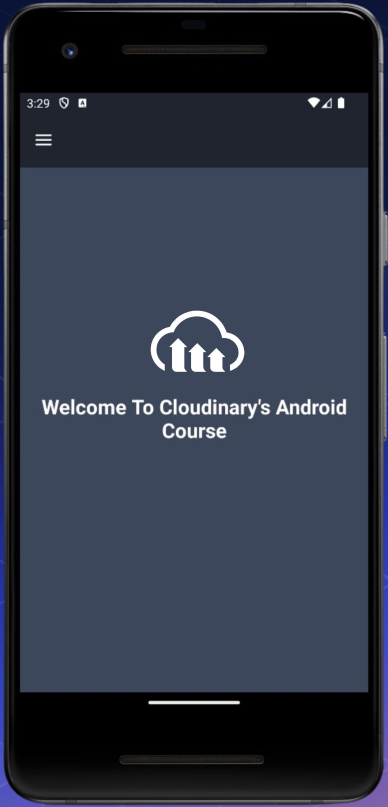
```
2024-02-06 15:11:42: Launching app on 'Pixel 2 API Test AVD'.
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.cloudinary.academy_course/.Splash.InitialActivity }

Open logcat panel for emulator Pixel 2 API Test AVD
Connected to process 5391 on device 'Pixel_2_API_Test_AVD [emulator-5554]'.
```



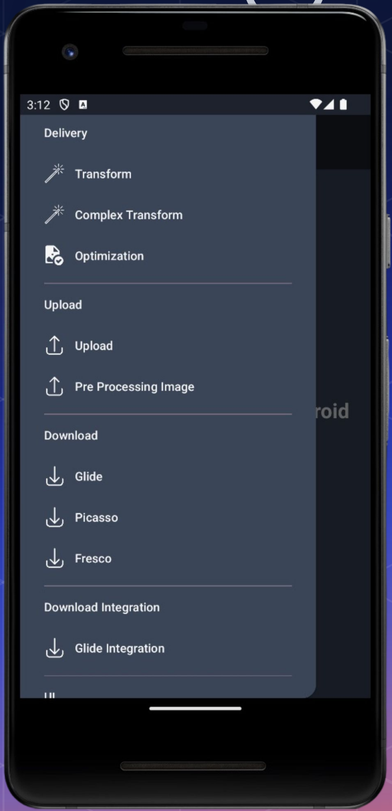
I have chosen the Google Pixel 2, but there are many devices to choose from.

Android Emulator

If you run the code you've cloned from the repository, this application will appear in the emulator and allow you to see the output of your code in real time.

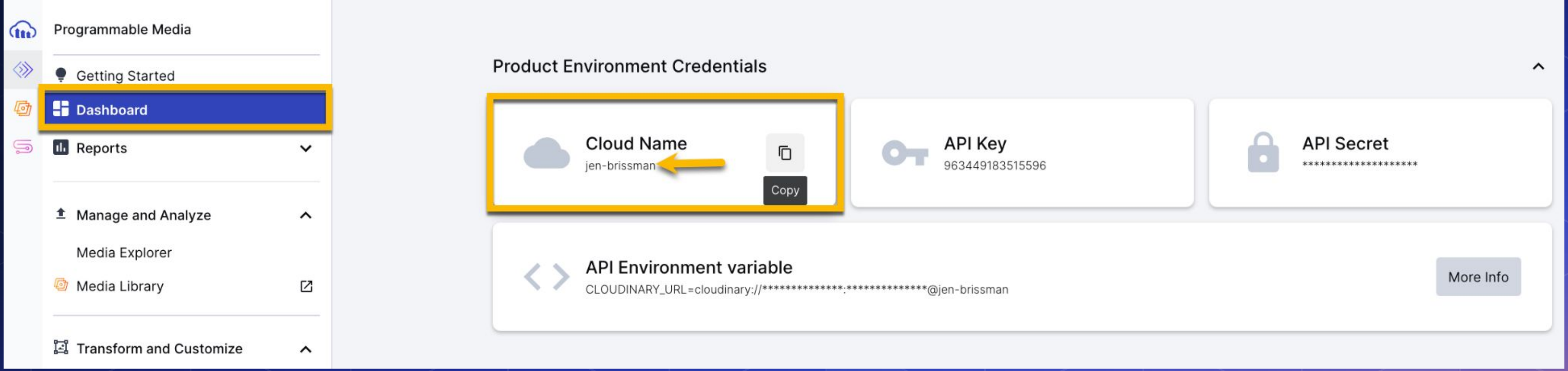


Welcome To Cloudinary's Android Course



Product Environment Credentials

You can locate your **Cloud Name** within your **Cloudinary Dashboard** under **Product Environment Credentials**.



Product Environment Credentials

Cloud Name
jen-brissman

API Key
963449183515596

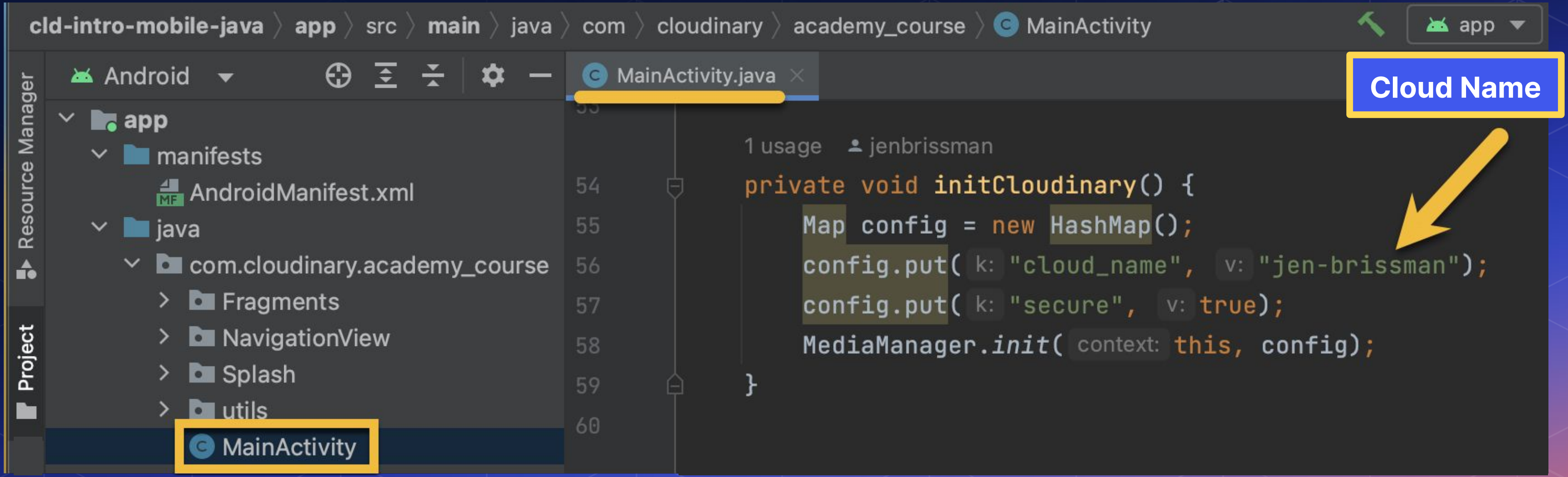
API Secret
****

API Environment variable
CLOUDINARY_URL=cloudinary://**************:**************@jen-brissman

Cloudinary

Configuring Your Environment

All you need to do is enter your cloud_name in the **MainActivity.java** file.



```
private void initCloudinary() {
    Map config = new HashMap();
    config.put( k: "cloud_name",  v: "jen-brissman");
    config.put( k: "secure",  v: true);
    MediaManager.init( context: this, config);
}
```

Upload

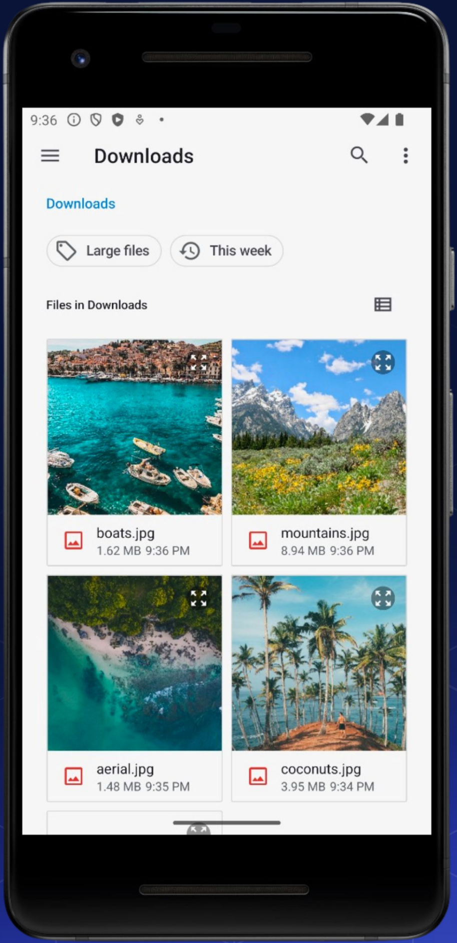
- Uploading an Asset
- Cloudinary Upload Widget
- Pre-Processing

Upload

Uploading media using Cloudinary's Android SDK is easy. Cloudinary streamlines the process for seamless integration into your Android app. In just a few lines of code, you can upload images and videos to Cloudinary.

```
private void uploadImage(Uri fileUri){
    String requestId = MediaManager.get().upload(fileUri)
    .unsigned("unsigned-image")
```

Cloudinary

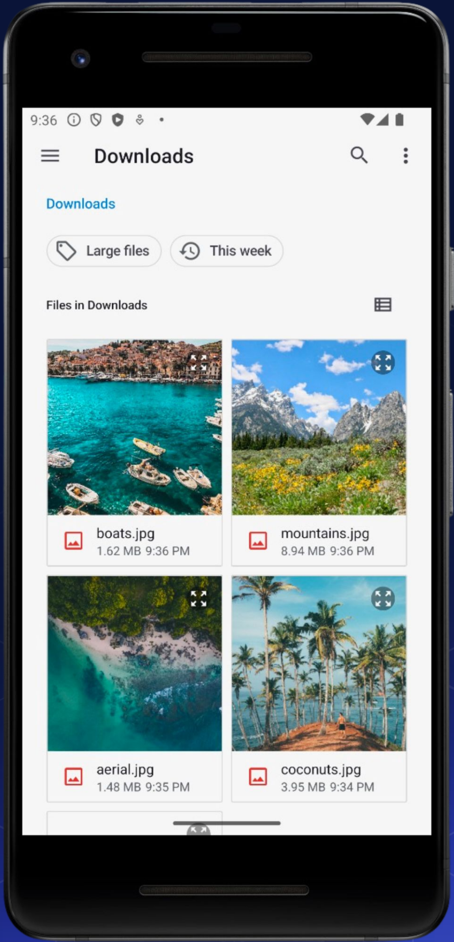


Upload Widget

You can implement Cloudinary's Upload Widget in the Android SDK to easily upload assets from the frontend directly to your Cloudinary account, without involving any servers in the process.

This ready-to-use widget offers developers a hassle-free integration, empowering users to effortlessly select and upload images or videos from their devices. This simplifies the user experience, enhances your app's media capabilities, and expedites development.

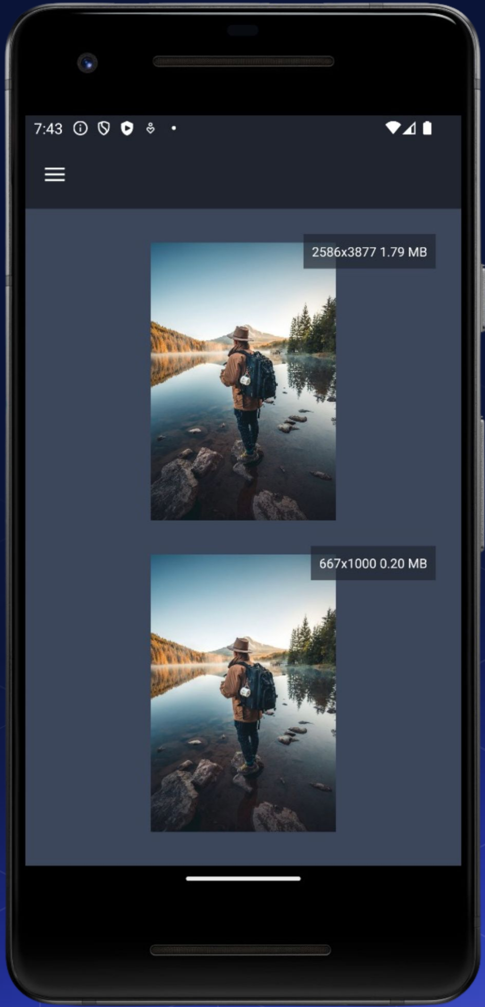
You can customize the widget to match your app's design all while benefiting from Cloudinary's powerful features.



Upload Widget - Code

```
private void setUploadButton() {
   Button uploadButton = binding.uploadWidgetButton;
   uploadButton.setOnClickListener(view -> openUploadWidget());

private void openUploadWidget() {
   UploadWidget.startActivity(getActivity(), UPLOAD_WIDGET_CODE);
   binding.uploadWidgetProgressbar.setVisibility(View.VISIBLE)}
```

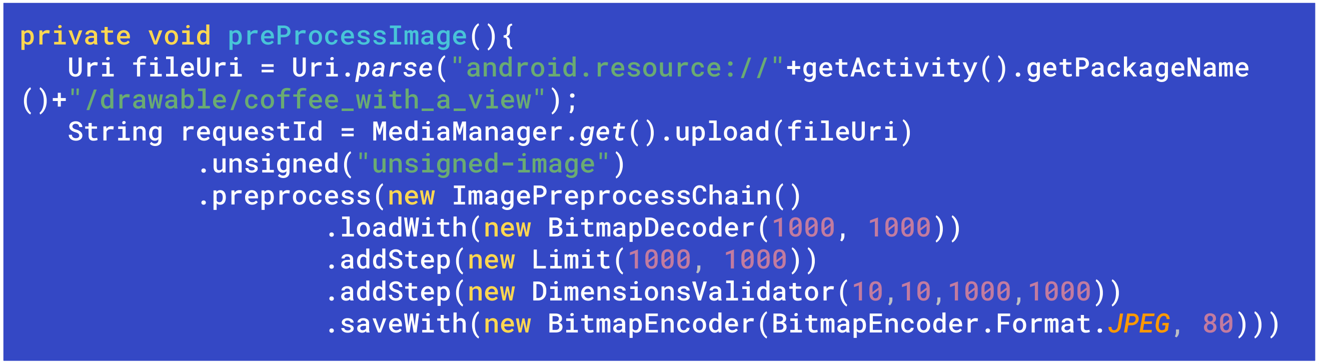
Cloudinary



Pre-Processing

Pre-processing allows you to perform image or video transformations **before** uploading them to the Cloudinary server.

Developers can manipulate and optimize media files on the client-side, such as resizing, cropping, adding effects, or adjusting quality, before transferring them to Cloudinary's cloud storage.

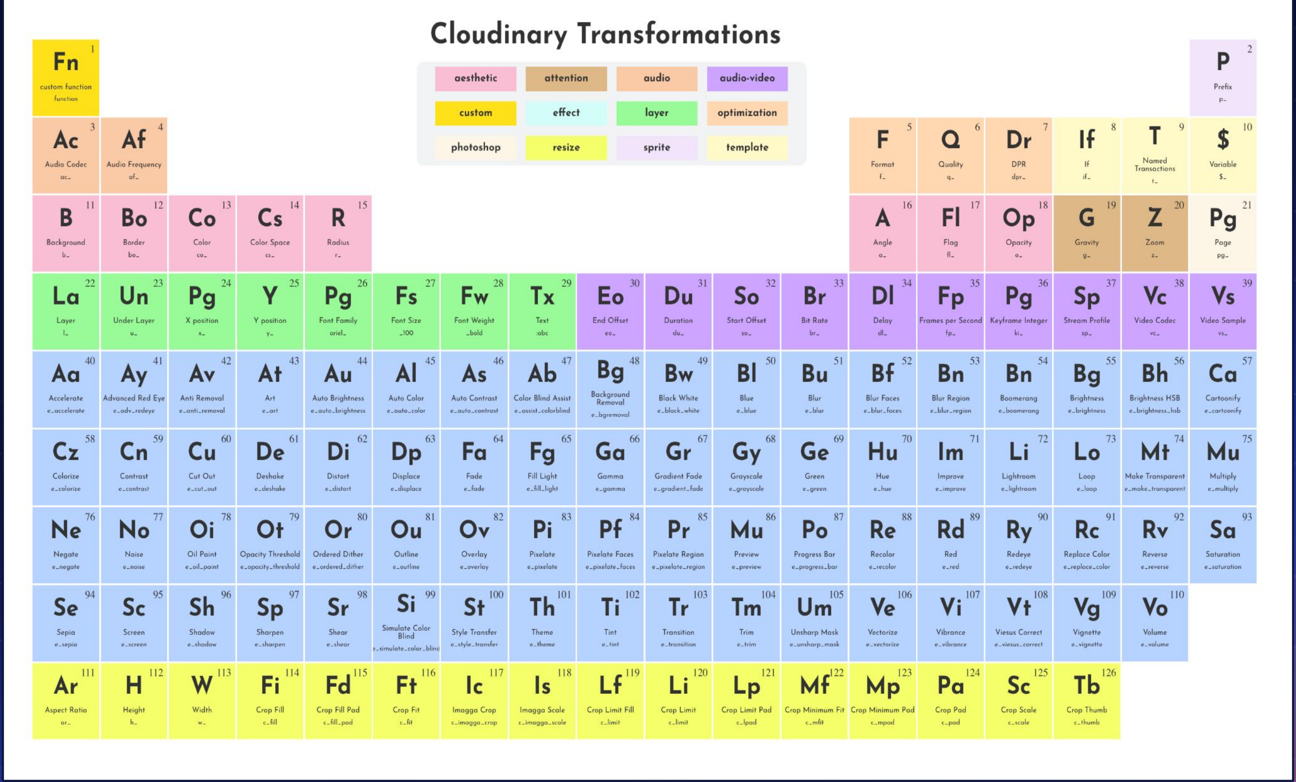


```
private void preProcessImage(){
   Uri fileUri = Uri.parse("android.resource://"+getActivity().getPackageName
()+"/drawable/coffee_with_a_view");
   String requestId = MediaManager.get().upload(fileUri)
           .unsigned("unsigned-image")
           .preprocess(new ImagePreprocessChain()
                   .loadWith(new BitmapDecoder(1000, 1000))
                   .addStep(new Limit(1000, 1000))
                   .addStep(new DimensionsValidator(10,10,1000,1000))
                   .saveWith(new BitmapEncoder(BitmapEncoder.Format.JPEG, 80)))
```

Asset Delivery

- Transformations
- Optimizations

Cloudinary Transformations

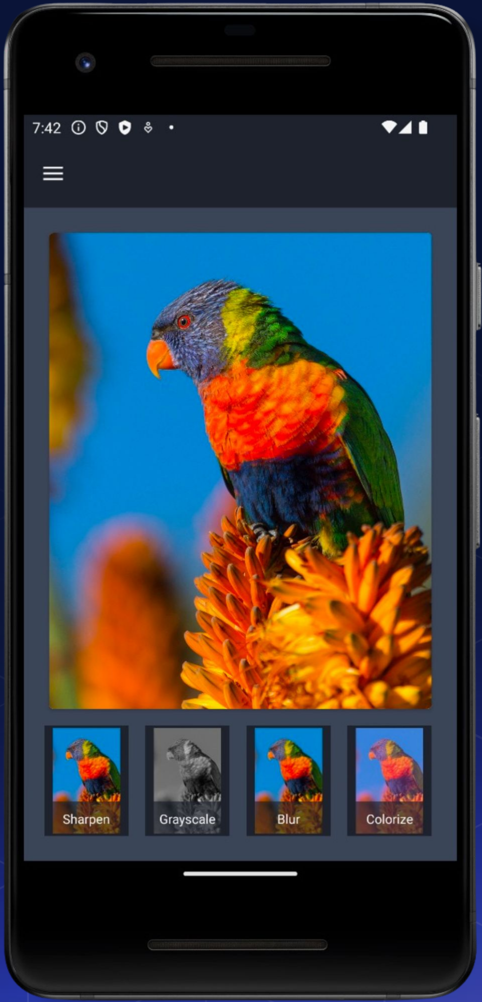


Cloudinary Transformations

aesthetic	attention	audio	audio-video
custom	effect	layer	optimization
photoshop	resize	sprite	template

1 Fn custom function function																	2 P Prefix p_
3 Ac Audio Codec ac_	4 Af Audio Frequency af_											5 F Format f_	6 Q Quality q_	7 Dr DPR dpr_	8 If If if_	9 T Named Transactions t_	10 $ Variable $_
11 B Background b_	12 Bo Border bo_	13 Co Color co_	14 Cs Color Space cs_	15 R Radius r_								16 A Angle a_	17 Fl Flag fl_	18 Op Opacity o_	19 G Gravity g_	20 Z Zoom z_	21 Pg Page pg_
22 La Layer l_	23 Un Under Layer u_	24 Pg X position x_	25 Y Y position y_	26 Pg Font Family ariel_	27 Fs Font Size _100	28 Fw Font Weight _bold	29 Tx Text :abc	30 Eo End Offset eo_	31 Du Duration du_	32 So Start Offset so_	33 Br Bit Rate br_	34 Dl Delay dl_	35 Fp Frames per Second fp_	36 Pg Keyframe Integer ki_	37 Sp Stream Profile sp_	38 Vc Video Codec vc_	39 Vs Video Sample vs_
40 Aa Accelerate e_accelerate	41 Ay Advanced Red Eye e_adv_redeye	42 Av Anti Removal e_anti_removal	43 At Art e_art	44 Au Auto Brightness e_auto_brightness	45 Al Auto Color e_auto_color	46 As Auto Contrast e_auto_contrast	47 Ab Color Blind Assist e_assist_colorblind	48 Bg Background Removal e_bgremoval	49 Bw Black White e_black_white	50 Bl Blue e_blue	51 Bu Blur e_blur	52 Bf Blur Faces e_blur_faces	53 Bn Blur Region e_blur_region	54 Bn Boomerang e_boomerang	55 Bg Brightness e_brightness	56 Bh Brightness HSB e_brightness_hsb	57 Ca Cartoonify e_cartoonify
58 Cz Colorize e_colorize	59 Cn Contrast e_contrast	60 Cu Cut Out e_cut_out	61 De Deshake e_deshake	62 Di Distort e_distort	63 Dp Displace e_displace	64 Fa Fade e_fade	65 Fg Fill Light e_fill_light	66 Ga Gamma e_gamma	67 Gr Gradient Fade e_gradient_fade	68 Gy Grayscale e_grayscale	69 Ge Green e_green	70 Hu Hue e_hue	71 Im Improve e_improve	72 Li Lightroom e_lightroom	73 Lo Loop e_loop	74 Mt Make Transparent e_make_transparent	75 Mu Multiply e_multiply
76 Ne Negate e_negate	77 No Noise e_noise	78 Oi Oil Paint e_oil_paint	79 Ot Opacity Threshold e_opacity_threshold	80 Or Ordered Dither e_ordered_dither	81 Ou Outline e_outline	82 Ov Overlay e_overlay	83 Pi Pixelate e_pixelate	84 Pf Pixelate Faces e_pixelate_faces	85 Pr Pixelate Region e_pixelate_region	86 Mu Preview e_preview	87 Po Progress Bar e_progress_bar	88 Re Recolor e_recolor	89 Rd Red e_red	90 Ry Redeye e_redeye	91 Rc Replace Color e_replace_color	92 Rv Reverse e_reverse	93 Sa Saturation e_saturation
94 Se Sepia e_sepia	95 Sc Screen e_screen	96 Sh Shadow e_shadow	97 Sp Sharpen e_sharpen	98 Sr Shear e_shear	99 Si Simulate Color Blind e_simulate_color_blind	100 St Style Transfer e_style_transfer	101 Th Theme e_theme	102 Ti Tint e_tint	103 Tr Transition e_transition	104 Tm Trim e_trim	105 Um Unsharp Mask e_unsharp_mask	106 Ve Vectorize e_vectorize	107 Vi Vibrance e_vibrance	108 Vt Viesus Correct e_viesus_correct	109 Vg Vignette e_vignette	110 Vo Volume e_volume	
111 Ar Aspect Ratio ar_	112 H Height h_	113 W Width w_	114 Fi Crop Fill c_fill	115 Fd Crop Fill Pad c_fill_pad	116 Ft Crop Fit c_fit	117 Ic Imagga Crop c_imagga_crop	118 Is Imagga Scale c_imagga_scale	119 Lf Crop Limit Fill c_limit	120 Li Crop Limit c_limit	121 Lp Crop Limit Pad c_lpad	122 Mf Crop Minimum Fit c_mfit	123 Mp Crop Minimum Pad c_mpad	124 Pa Crop Pad c_pad	125 Sc Crop Scale c_scale	126 Tb Crop Thumb c_thumb		

Cloudinary



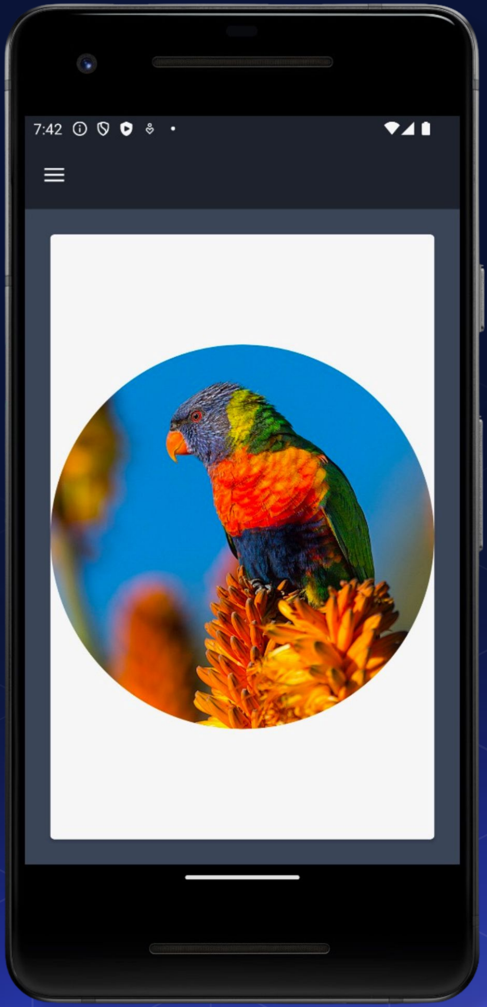
Transformations

Cloudinary allows you to seamlessly integrate transformations into your workflow. These transformation capabilities include resizing, cropping, format conversion, and other advanced effects.

This enhances user experiences, improves performance, and delivers streamlined media assets at scale across the web.

Transformations - Code

```
private List<String> buildTransformUrls() {
   String lorikeetOriginalUrl = MediaManager.get().url().generate("lorikeet");
   String lorikeetSharpenUrl = MediaManager.get().url().transformation(new
Transformation().effect("sharpen:400")).generate("lorikeet");
   String lorikeetGrayscaleUrl = MediaManager.get().url().transformation(new
Transformation().effect("grayscale")).generate("lorikeet");
   String lorikeetBlurUrl = MediaManager.get().url().transformation(new
Transformation().effect("blur:500")).generate("lorikeet");
   String lorikeetColorizeUrl = MediaManager.get().url().transformation(new
Transformation().color("violet").effect("colorize:30")).generate("lorikeet");
   return Arrays.asList(lorikeetOriginalUrl, lorikeetSharpenUrl,
lorikeetGrayscaleUrl, lorikeetBlurUrl, lorikeetColorizeUrl);}
```



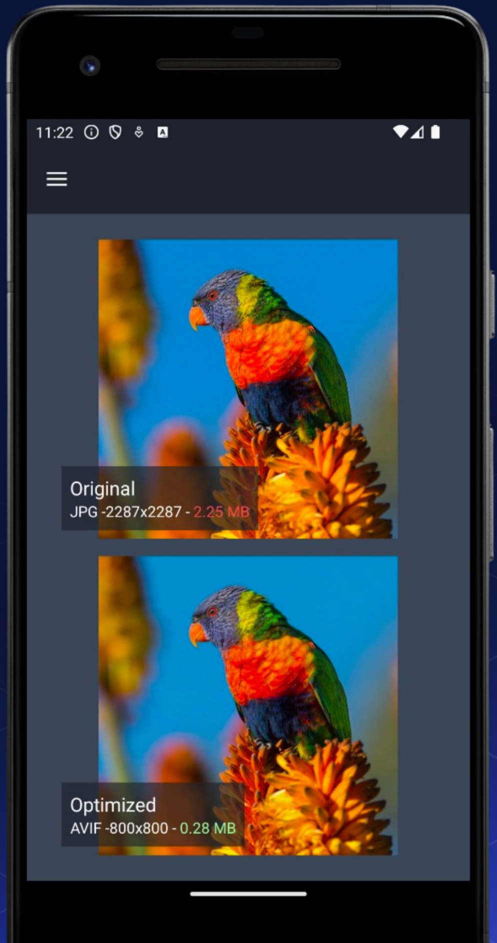
Complex Transformations

Cloudinary allows you to combine multiple transformation features to customize your media assets precisely to your needs.

Whether resizing, cropping, adding effects, or converting formats, Cloudinary simplifies the process, allowing you to seamlessly integrate and apply multiple transformations in the same call.

Complex Transformations - Code

```java
public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {
   super.onViewCreated(view, savedInstanceState);
   String url = MediaManager.get().url().transformation(new
Transformation().gravity("auto").effect("sharpen:400").radius("max")
.fetchFormat("auto").quality("auto")).generate("lorikeet");
   setMainImageView(url);}
```



Original
JPG -2287x2287 - 2.25 MB

Optimized
AVIF -800x800 - 0.28 MB

No visual difference

Optimization

Cloudinary allows you to enhance your images and videos for optimal performance and user experience with just a few simple steps.

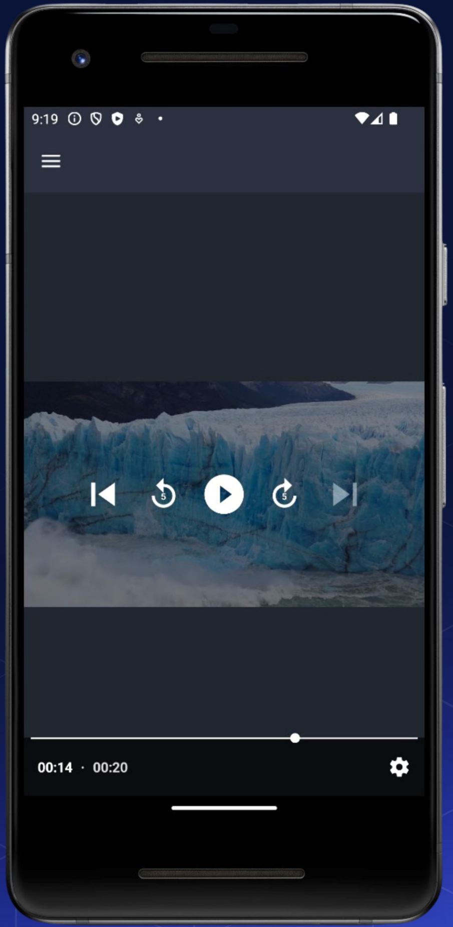
From automatic format selection **(f_auto)** and quality adjustment **(q_auto)** to intelligent resizing and compression, Cloudinary's optimization tools are designed to streamline your workflow and improve load times across all devices and platforms.

Optimization - Code

```
private void setOptimizationImageView((String publicId)){
   String url = MediaManager.get().url().transformation(new
Transformation().crop("scale").width(800).fetchFormat("avif").quality("auto")
.dpr("auto")).generate(publicId);
```

Video

- Cloudinary Native Video Player
- Creating a Video Feed



CldVideoPlayer

The Cloudinary Video Player, allows developers to easily embed and customize video playback experiences within their Android apps.

The player supports various video formats and codecs, ensuring compatibility with a wide range of devices and platforms.

Additionally, it offers advanced features such as adaptive bitrate streaming, allowing for smooth playback even under varying network conditions.

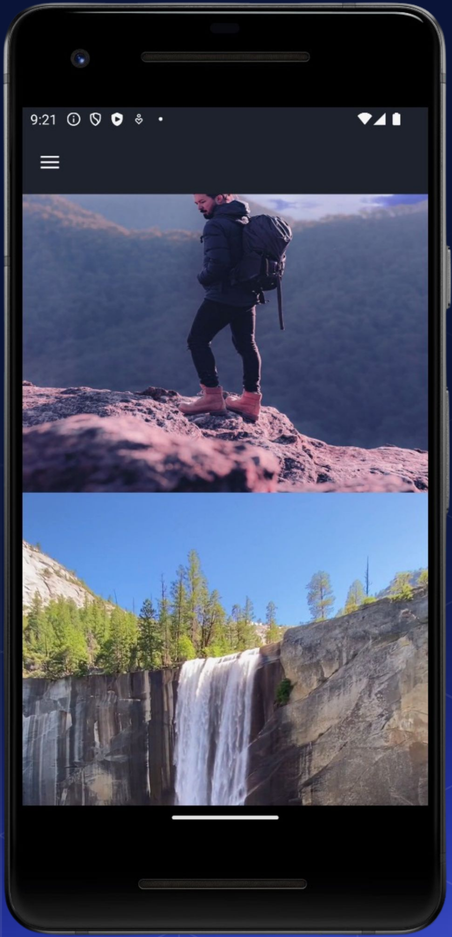
Cloudinary

CldVideoPlayer - Code

```
private void setVideoPlayer() {
   CldVideoPlayer player = null;
   player = new CldVideoPlayer(getContext(),
           (MediaManager.get().url().resourceType("video")
.transformation(new
Transformation<>().quality("auto")).generate("glacier")));
   binding.playerView.setPlayer(player.getPlayer());
   player.getPlayer().play();}
```

CldVideoPlayer is a wrapper around ExoPlayer, an open-source media playback library for Android developed by Google. To get the video player, we need to call getPlayer().

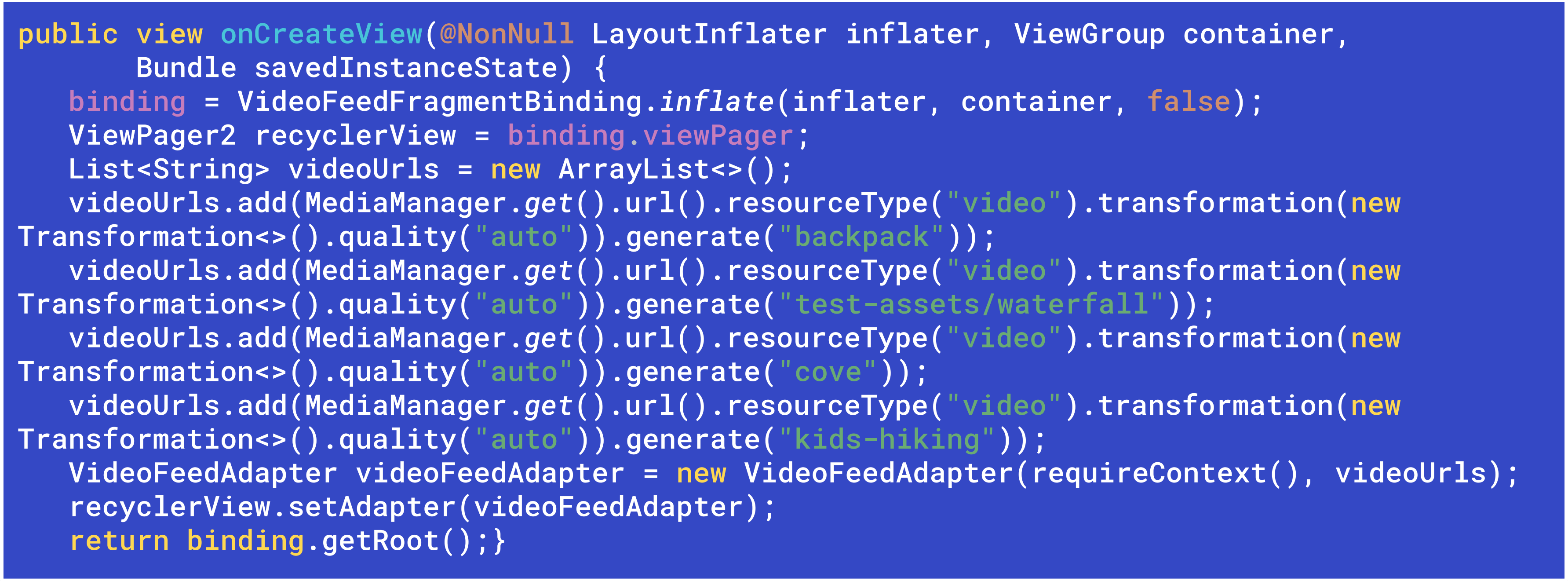
Cloudinary



Video Feed

You can also create a video feed using the Android SDK.

Video Feed - Code



```
public view onCreateView(@NonNull LayoutInflater inflater, ViewGroup container,
       Bundle savedInstanceState) {
   binding = VideoFeedFragmentBinding.inflate(inflater, container, false);
   ViewPager2 recyclerView = binding.viewPager;
   List<String> videoUrls = new ArrayList<>();
   videoUrls.add(MediaManager.get().url().resourceType("video").transformation(new
Transformation<>().quality("auto")).generate("backpack"));
   videoUrls.add(MediaManager.get().url().resourceType("video").transformation(new
Transformation<>().quality("auto")).generate("test-assets/waterfall"));
   videoUrls.add(MediaManager.get().url().resourceType("video").transformation(new
Transformation<>().quality("auto")).generate("cove"));
   videoUrls.add(MediaManager.get().url().resourceType("video").transformation(new
Transformation<>().quality("auto")).generate("kids-hiking"));
   VideoFeedAdapter videoFeedAdapter = new VideoFeedAdapter(requireContext(), videoUrls);
   recyclerView.setAdapter(videoFeedAdapter);
   return binding.getRoot();}
```

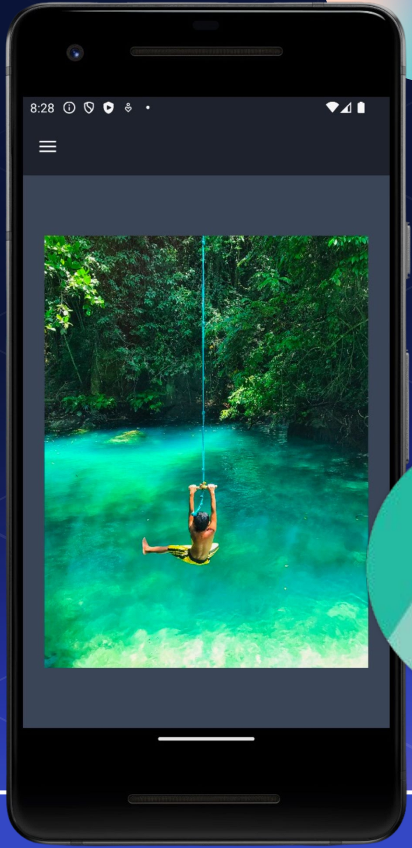
Download - Image Loaders

- Glide
- Picasso
- Fresco

Glide Download

Glide for Android is a powerful and flexible open-source image loading and caching library developed by the Google engineering team.

Glide's robust caching mechanisms are designed to enhance the performance and efficiency of image loading tasks in your Android application, with support for fetching images from various sources like URLs, local files, and content providers.

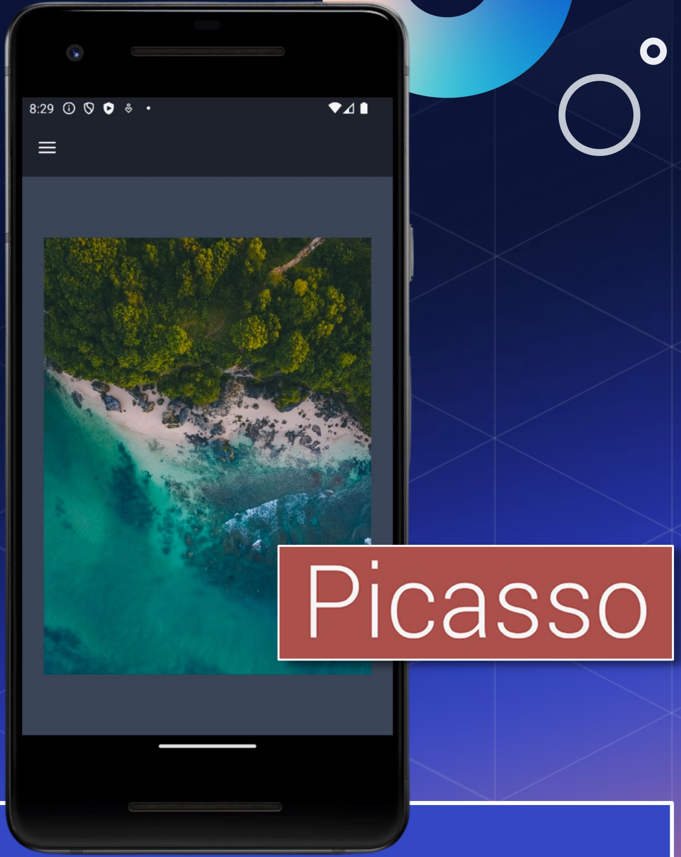


```
private void setImageViewWithGlideIntegration() {

MediaManager.get().setDownloadRequestBuilderFactory(new GlideDownloadRequestBuilderFactory());
ImageView imageView = binding.glideDownloadImageview;
MediaManager.get().download(getActivity()).load(MediaManager.get().url().generate("swing")).into(imageView);}
```

Picasso Download

Picasso for Android is an open-source image loading library developed by Square.

Picasso provides features such as image resizing, caching, and transformation, making it a popular choice among Android developers for efficiently handling image loading tasks.

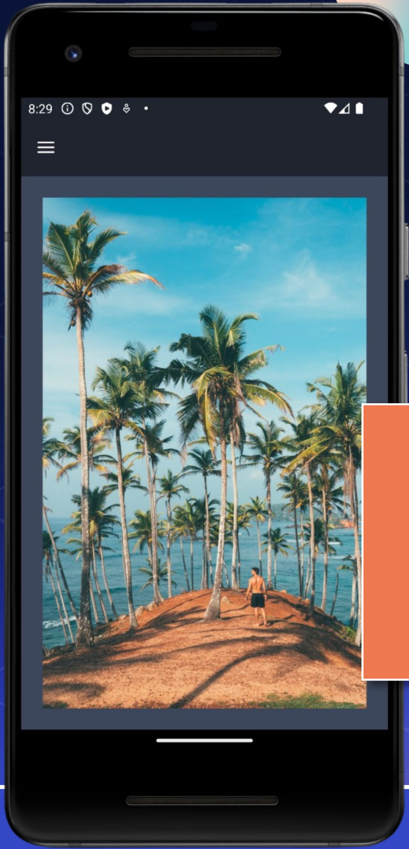


```
private void setImageViewWithPicassoIntegration() {

MediaManager.get().setDownloadRequestBuilderFactory(new PicassoDownloadRequestBuilderFactory());
ImageView imageView = binding.picassoDownloadImageview;
MediaManager.get().download(getActivity()).load(MediaManager.get().url().generate("aerial")).into(imageView);}
```

Fresco Download

Fresco for Android is an open-source image loading library developed and maintained by Facebook.

Fresco offers features such as memory management, image caching, progressive image loading, animated GIF support, and more. It enhances the image loading and display experience within Android apps.



```
private void setImageViewWithFrescoIntegration() {

MediaManager.get().setDownloadRequestBuilderFactory(new FrescoDownloadRequestBuilderFactory());
ImageView imageView = binding.frescoDownloadImageview;
MediaManager.get().download(getActivity()).load(MediaManager.get().url().generate("coconuts")).into(imageView);}
```

Integration

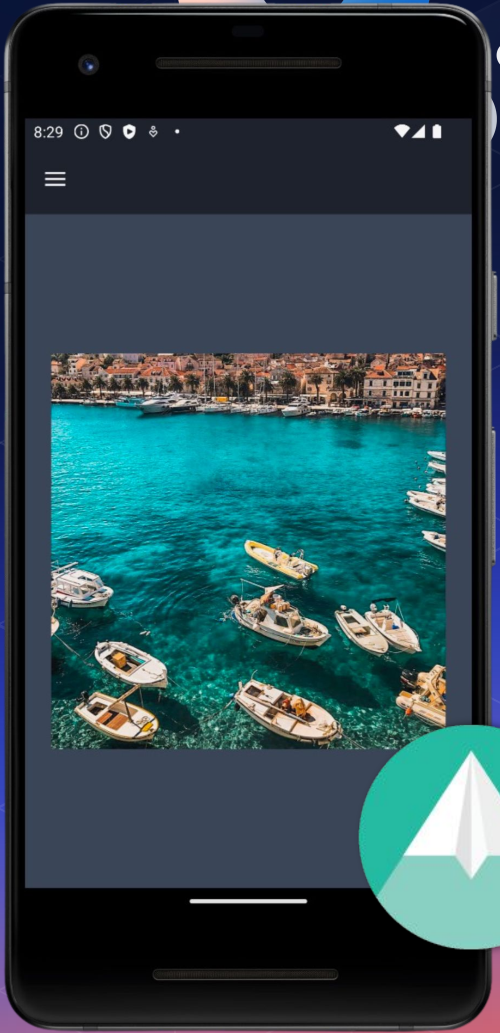
- Glide Integration

Glide Integration

Integrating Cloudinary with Glide, allows developers to easily fetch, transform, and display images and videos from Cloudinary's cloud storage while benefiting from Glide's efficient caching and image loading capabilities.

This integration streamlines the process of fetching and displaying media assets in Android apps, while also providing the flexibility to apply dynamic transformations and optimizations to images and videos on-the-fly, based on the specific requirements of the application.

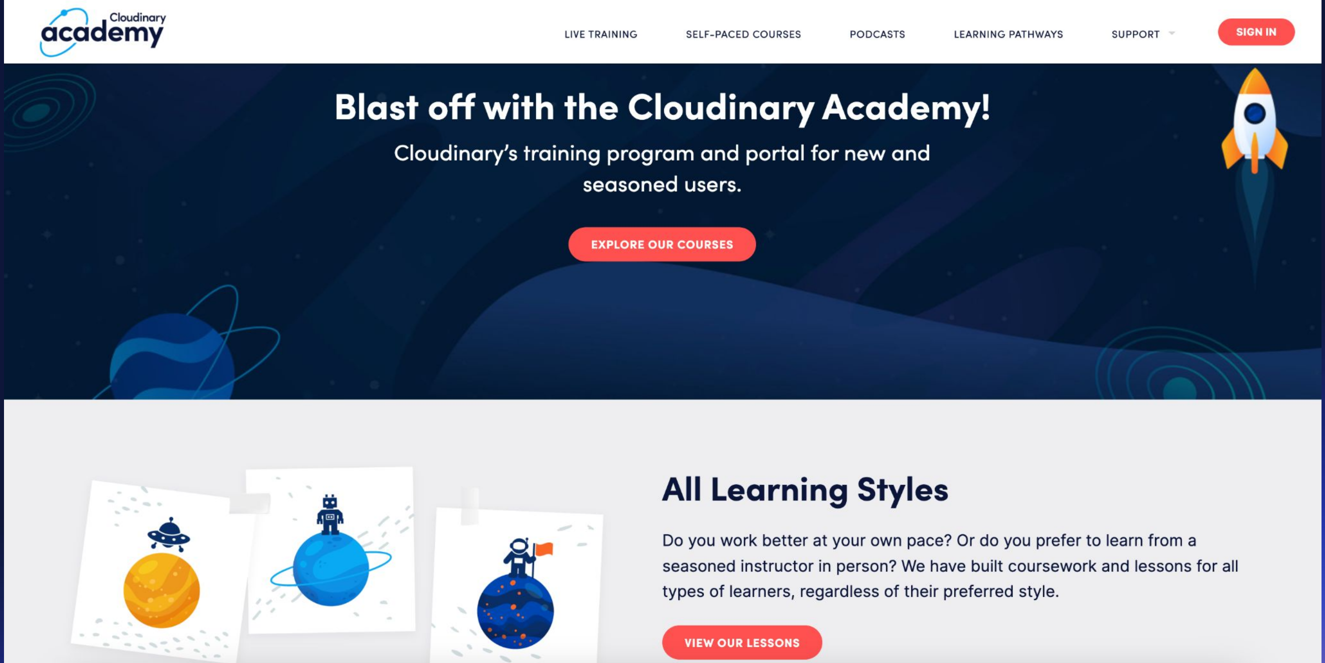
```
private void setImageViewWithGlideIntegration() {
   ImageView imageView = binding.glideIntegrationImageview;
   GlideApp.with(imageView)
           .load(new CloudinaryRequest.Builder("boats")
                   .transformation(new
Transformation().effect("blur"))
                   .responsive(ResponsiveUrl.Preset.AUTO_FILL)
                   .build())
           .into(imageView);}
```



Next Steps and Further Support

Cloudinary Academy

The Cloudinary Academy offers both self-paced courses and live classes taught by Cloudinary experts.



Blast off with the Cloudinary Academy!

Cloudinary's training program and portal for new and seasoned users.

All Learning Styles

Do you work better at your own pace? Or do you prefer to learn from a seasoned instructor in person? We have built coursework and lessons for all types of learners, regardless of their preferred style.

training.cloudinary.com

Cloudinary

Recommended Courses

Those who are new to Cloudinary's APIs can benefit from a variety of helpful, self-paced courses that provide comprehensive learning resources.

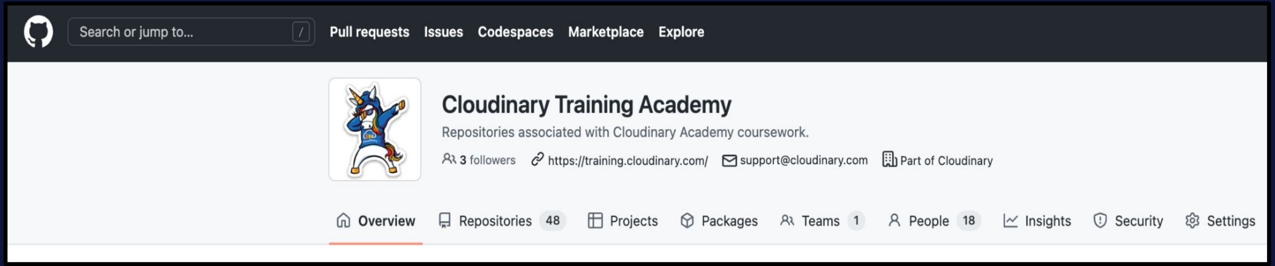
- Introduction to Cloudinary Programmable Media (90-Minutes)
- Understanding Cloudinary Programmable Media Terminology (30-Minutes)
- Cloudinary JumpStart for New Developer Users (~40-Minutes)
- Advanced Concepts for Developers (~16 hours)
- Introduction for Node.js Developers (90 minutes)
- Fundamentals for Developers (~9 hours)

Cloudinary academy

training.cloudinary.com

Review our GitHub Repositories

Access our sample projects to help you with building your own work using Cloudinary.

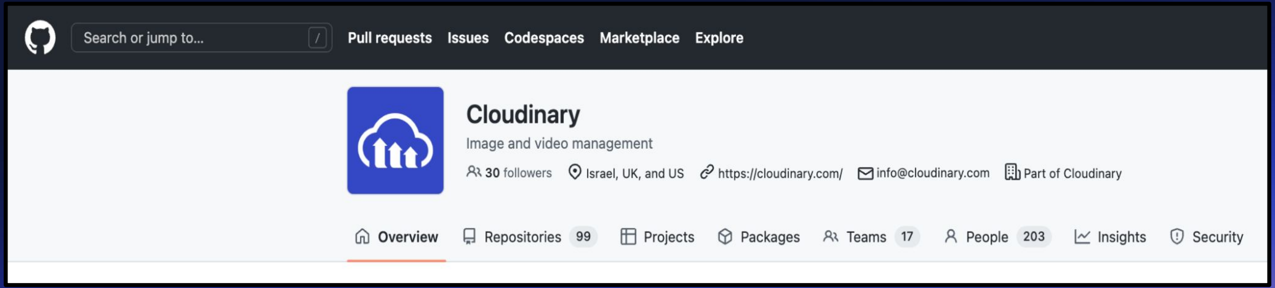


Cloudinary Training Academy

Repositories associated with Cloudinary Academy coursework.

3 followers https://training.cloudinary.com/ support@cloudinary.com Part of Cloudinary

github.com/cloudinary-training



Cloudinary

Image and video management

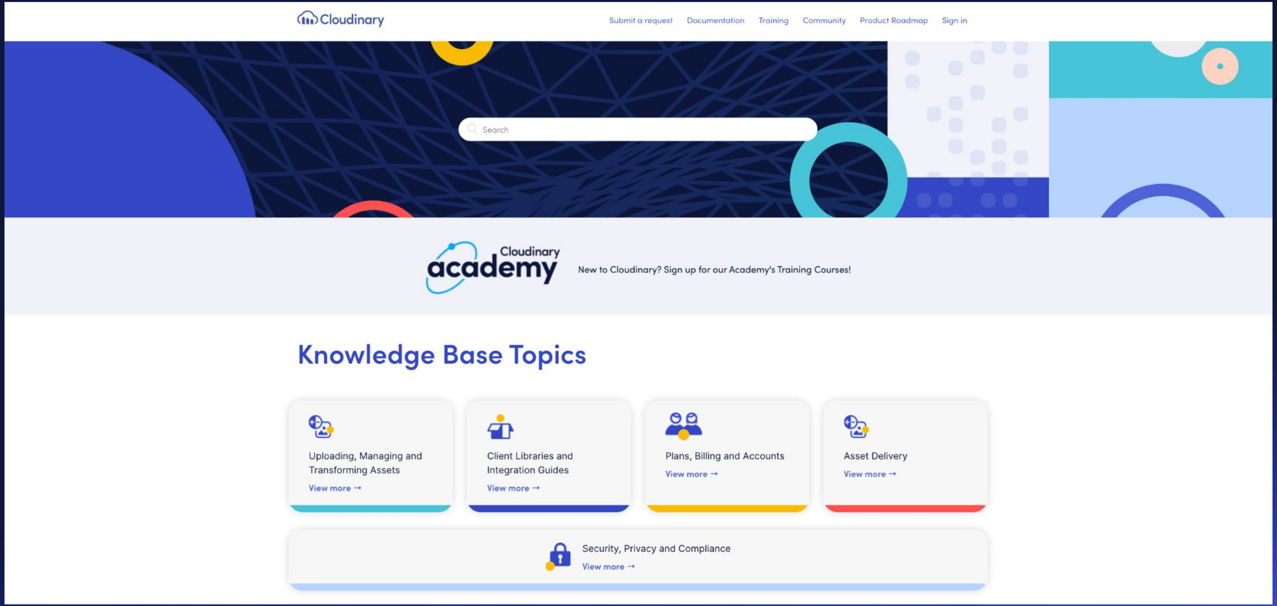
30 followers Israel, UK, and US https://cloudinary.com/ info@cloudinary.com Part of Cloudinary

github.com/cloudinary

Cloudinary

Engineering Support

We are always happy to answer your questions, as we have dedicated support staff for our developer community.



Cloudinary

Submit a request | Documentation | Training | Community | Product Roadmap | Sign in

Search

Cloudinary academy

New to Cloudinary? Sign up for our Academy's Training Courses!

Knowledge Base Topics

Uploading, Managing and Transforming Assets
View more →

Client Libraries and Integration Guides
View more →

Plans, Billing and Accounts
View more →

Asset Delivery
View more →

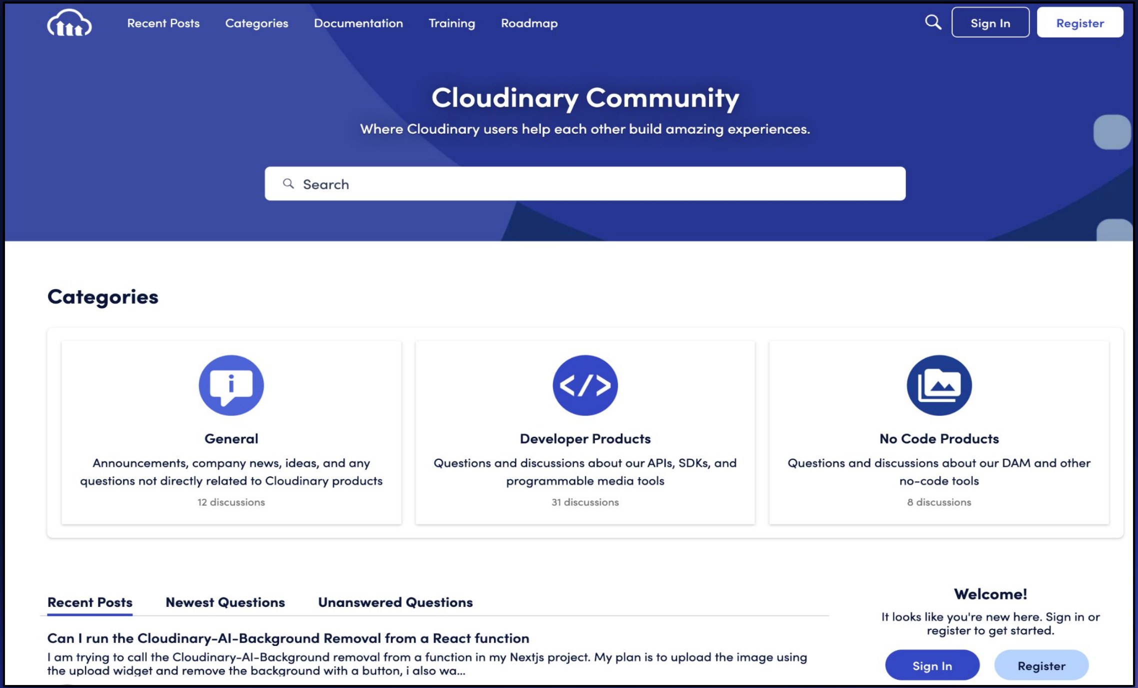
Security, Privacy and Compliance
View more →

support.cloudinary.com/hc/en-us/requests/new

Cloudinary

Join Our Community Forums

Ask questions to staff or other users in our dedicated communities.



Cloudinary Community

Where Cloudinary users help each other build amazing experiences.

Search

Categories

General
Announcements, company news, ideas, and any questions not directly related to Cloudinary products
12 discussions

Developer Products
Questions and discussions about our APIs, SDKs, and programmable media tools
31 discussions

No Code Products
Questions and discussions about our DAM and other no-code tools
8 discussions

Recent Posts | Newest Questions | Unanswered Questions

Can I run the Cloudinary-AI-Background Removal from a React function
I am trying to call the Cloudinary-AI-Background removal from a function in my Nextjs project. My plan is to upload the image using the upload widget and remove the background with a button, i also wa...

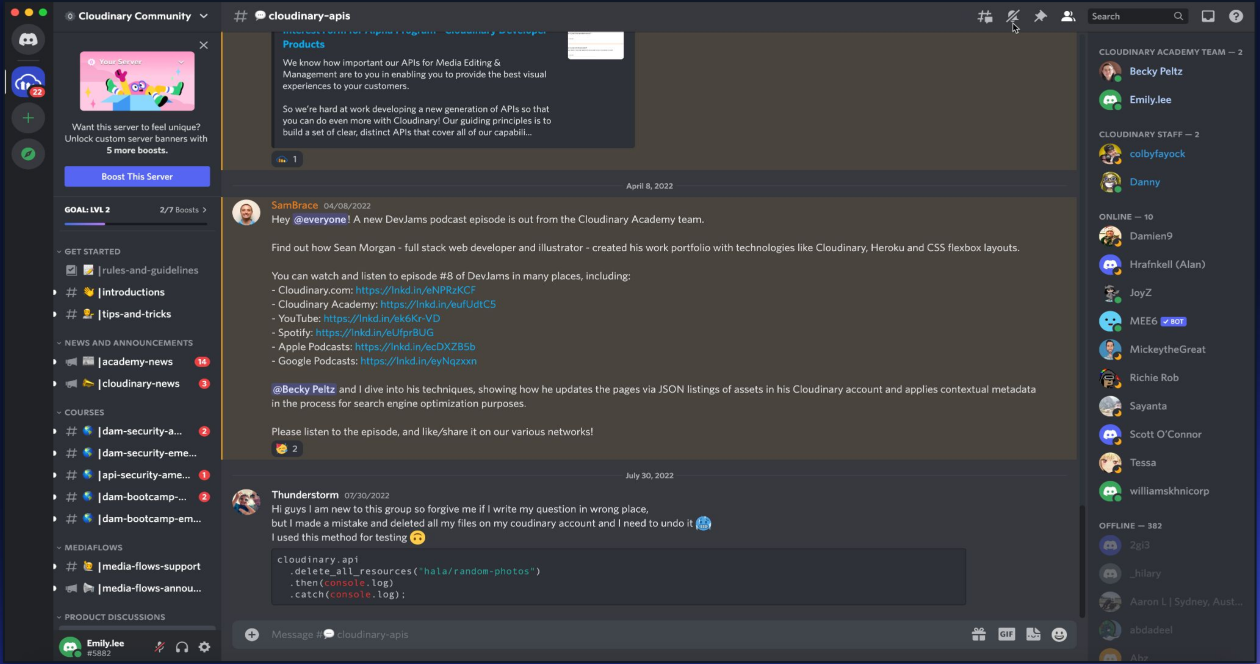
Welcome!
It looks like you're new here. Sign in or register to get started.

community.cloudinary.com

Cloudinary

Join Our Discord Discussions

Connect with Cloudinary users and staff on Discord!



Products

We know how important our APIs for Media Editing & Management are to you in enabling you to provide the best visual experiences to your customers.

So we're hard at work developing a new generation of APIs so that you can do even more with Cloudinary! Our guiding principles is to build a set of clear, distinct APIs that cover all of our capabili...

1

April 8, 2022

SamBrace 04/08/2022

Hey @everyone! A new DevJams podcast episode is out from the Cloudinary Academy team.

Find out how Sean Morgan - full stack web developer and illustrator - created his work portfolio with technologies like Cloudinary, Heroku and CSS flexbox layouts.

You can watch and listen to episode #8 of DevJams in many places, including:
- Cloudinary.com: https://lnkd.in/eNPRzKCF
- Cloudinary Academy: https://lnkd.in/eufUdtC5
- YouTube: https://lnkd.in/ek6Kr-VD
- Spotify: https://lnkd.in/eUfprBUG
- Apple Podcasts: https://lnkd.in/ecDXZB5b
- Google Podcasts: https://lnkd.in/eyNqzxxn

@Becky Peltz and I dive into his techniques, showing how he updates the pages via JSON listings of assets in his Cloudinary account and applies contextual metadata in the process for search engine optimization purposes.

Please listen to the episode, and like/share it on our various networks!

2

July 30, 2022

Thunderstorm 07/30/2022

Hi guys I am new to this group so forgive me if I write my question in wrong place,
but I made a mistake and deleted all my files on my coudinary account and I need to undo it 🥶
I used this method for testing 🙃

```
cloudinary.api
  .delete_all_resources("hala/random-photos")
  .then(console.log)
  .catch(console.log);
```

discord.gg/cloudinary

Cloudinary

Thank you