

Cloudinary upload/destroy via API

1-Prepare

Create an account

Here -> <https://cloudinary.com/>

Add bundle

```
composer require cloudinary/cloudinary_php
```

More infos https://cloudinary.com/documentation/php_integration/

Add vars in .env with your cloudinary account infos from dashboard

```
CLOUDINARY_URL=cloudinary://yourApiKey:yourApiSecret@yourCloudName  
CLOUDINARY_NAME="yourCloudName"  
CLOUDINARY_API_KEY="yourApiKey"  
CLOUDINARY_API_SECRET="yourApiSecret"
```

2-Add Service

```

<?php
//src/Service/ImageUploader.php

namespace App\Service;

use Cloudinary\Api\ApiResponse;
use Cloudinary\Api\Upload\UploadApi;
use Cloudinary\Configuration\Configuration;

class ImageUploader
{

    public function __construct()
    {
        Configuration::instance([
            "cloud_name" => $_ENV['CLOUDINARY_NAME'],
            "api_key" => $_ENV['CLOUDINARY_API_KEY'],
            "api_secret" => $_ENV['CLOUDINARY_API_SECRET'],
            "secure" => FALSE]);
    }

    public function upload($file, $options = []): ApiResponse
    {
        $response=(new UploadApi())->upload( $file, $options);
        return $response;
    }

    public function remove($file, $options = []): ApiResponse
    {
        $response=(new UploadApi())->destroy( $file, $options);
        return $response;
    }
}

```

3-Controller

```

...
use App\Service\ImageUploader; // the service
use Gedmo\Sluggable\Util\Urlizer; //the urlizer from gedmo bundle
use Symfony\Component\Filesystem\Filesystem; //to move et remove files
...

```

To upload:

```

#[Route('/new', name: 'app_photo_chambre_new', methods: ['GET', 'POST'])]
public function new(Request $request, PhotoRepository $photoRepository, $chambre, ChambresRepository $chambresRepository)
{
    if ($chambresRepository->find($chambre)->getHotel()->getGerant() !== $this->getUser()) {
        throw $this->createAccessDeniedException();
    }
    $photos= [];
    $form = $this->createForm(PhotoChambreType::class, $photos);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        $attachments = $form['images']->getData();
        if ($attachments) {
            foreach($attachments as $uploadedFile)
            {
                $photo = new Photo();
                /** @var UploadedFile $uploadedFile */
                $destination = $this->getParameter('kernel.project_dir').'/public/uploads/photos';
                $originalFilename = pathinfo($uploadedFile->getClientOriginalName(), PATHINFO_FILENAME);
                $newFilename = Urlizer::urlize($originalFilename).'-'.uniqid();
                $newFileExt = '.'.$uploadedFile->guessExtension();
                //upload image to public
                $uploadedFile->move(
                    $destination,
                    $newFilename . $newFileExt,
                    0777
                );
                //move image from public to cloudinary
                (new ImageUploader())->upload( $destination.'/'.$newFilename.$newFileExt,["public"]);
                //add image infos in database
                /** @var Chambres $chambres */
                $chambres = $chambresRepository->find($chambre);
                $photo->setChambres($chambres)
                    ->setCover(false)
                    ->setLien($newFilename.$newFileExt);
                $photoRepository->add($photo);
                //show message
                $this->addFlash('success', 'Votre image a été enregistrée');
                //delete image from public folder
                $filesystem = new Filesystem();
                $filesystem->remove($destination.'/'.$newFilename.$newFileExt);
            }
        }
        return $this->redirectToRoute('app_photo_chambre_index', [
            'chambre' => $chambre,
        ], Response::HTTP_SEE_OTHER);
    }
}

```

To delete:

```
#[Route('/{id}', name: 'app_photo_chambre_delete', methods: ['POST'])]
public function delete(Request $request, Photo $photo, PhotoRepository $photoRepository, $chambre,
{
    if ($chambresRepository->find($chambre)->getHotel()->getGerant() !== $this->getUser()) {
        throw $this->createAccessDeniedException();
    }
    if ($this->isCsrfTokenValid('delete'.$photo->getId(), $request->request->get('_token'))) {
        //get clouidinaryId
        $path = $photo->getLien();
        $cloudinaryId = substr($path, 0, strrpos($path, "."));
        //delete it from clouidinary and get result
        $result = (new ImageUploader())->remove( $cloudinaryId);
        if($result['result']=='ok'){
            //show message
            $this->addFlash('success', 'Votre image a été supprimée.' (.'$cloudinaryId.'));
            //delete from database
            $photoRepository->remove($photo);
        }
    }

    return $this->redirectToRoute('app_photo_chambre_index', [
        'photo' => $photo,
        'chambre' => $chambre,
    ], Response::HTTP_SEE_OTHER);
}
```