

Django 基于腾讯云对象存储 SDK 上传图片

作者:zyk

- 原文链接: https://ld246.com/article/1566965564833
- 来源网站:链滴
- 许可协议:署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

1. 前言

本篇文章将讲述如何利用腾讯云对象存储的python SDK上传图片,包含基础配置,原理概述和前后 搭建。

文章内容总共分为四个部分,一为环境准备,如何搭建云对象存储基于python的SDK环境;二为编后台,如何调用SDK中的上传图片函数,并对图像进行进一步的处理;三为前端交互,如何利用ajax装formData并提交至后台;最后对结果进行测试。

2. 环境准备

● 首先进入腾讯云控制台,进入对象存储管理中心,点击存储桶->创建存储桶,填写名称和编辑访问限。如果是搭建图床,权限可以设置为**共有读私有写**,因为图床一般需要给网站浏览者访问,但只有理员能上传图片。

	blog-img		-1300089264 🤅	\odot			
	仅支持小写字母、	数字和 - 的组合	,不能超过40字符				
所臞地域	中国	▼ 成都	*				
	与相同地域其他勝	讯云服务内网旦	〔通,创建后不可更改	文地域			
访问权限	○ 私有读写 (○ 公有读私有	写 🗌 公有读写				
	可对object进行置领	名读操作,写操作	F需要进行身份验证。				
	注意: 公有读权限可以 据安全,不推荐	以通过匿名身份 萨此配置,建议	直接读取您存储稀中 您选择私有。	的数据,存	在一定的安全风景	金,为确保您的数	
请求域名	注意: 公有读权限可以 据安全,不推获 blog-img-1300089 创建完成后,您可	以通过懂名身份 荐此配置,建议 264.cos.ap-che 以使用该域名对	直接读取您存储桶中 您选择私有。 engdu.myqcloud.com 存储桶进行访问	e的数据,存	在一定的安全风飘	金,为确保您的数	

• 创建完成以后,点击已经创建的存储桶,然后进入权限管理。

	× 🙆 存储场-控制台	× +				- o ×
← → C	loud.tencent.com/cos5/bucket/ ॥ 🤒 क्षार 🤒 संस्थ	setting?type=aclconfig&buck 第 🦲 VPN 🔮 首東 🔒 git	etName-zyk-1300089264	8:path=%252F8:region=ap-shanç Na 🔝 III.A:#BORUTOXI 💽 (yhal 如政 Bing 授家 🛛 😽 和松姑叔长工具	☆ 😢 : 大全 📄 東版ATT/s9+副町 =
🙆 🚓 🕁 .	RIMER +				2 acaes	t• mal• Im ≜® O ⊙
対象存储 111 概約	← zyk-1300089264 文件列表 基础配置	域名管理 权限管理	-			REACTER REACTER REACTER
 (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)		SAGANA CANES				
@ IN ·	用や実型	肥谷口 ①		62.98		授作
i (i) directifik (c	1231-15	100011220193		完全控制 成如而中		
	Policy权限设置					
	医形说道 策略道法					
	致力 用户类型	账档ID	的现在词	國民國作	四股部件	操作
	北许 所有用户	a -	zyk-1300089254/*	将和19位 1月11日	91	B139
5						

● Policy权限设置中删除默认的策略,然后添加策略,设置用户为 所有用户,操作为所有操作,不置可能会导致之后上传图片报403错误,然后保存即可。

*	用白米用	w Bio	10.0-
	用小茶菜	UIP M	15R1 F
	所有用户	(*)	删除
/		添加用户	
	○ 整个存储桶 │ 指定资源		
" 路径"	● 整个存储桶 指定资源 zyk-1300089264/*		
" 路径,	● 整个存储桶 / 指定资源 zyk-1300089264/*		
* 路径 *	 ● 整个存储桶 ○ 指定资源 zyk-1300089264/* 操作名称 		操作
• 器径•	 ● 整个存储桶 ● 指定资源 zyk-1300089264/* 操作名称 所有操作 ▼ 		操作
· 路径,	 ● 整个存储桶 ● 指定资源 zyk-1300089264/* 操作名称 所有操作 ▼ 		操作
。 路径,	 ● 整个存储桶 ● 指定资源 zyk-1300089264/* 操作名称 所有操作 ▼ 	添加操作	操作 删除
· 路径,	 ● 整个存储桶 ○ 指定资源 zyk-1300089264/* 操作名称 所有操作 ▼ 	添加操作	操作

● 由于使用了腾讯云对象存储的Python SDK,所以我们需要利用pip安装所需的第三方库。在命令行 输入 pip install -U cos-python-sdk-v5安装SDK,到此环境配置就结束了。

3. 编写后台

初始化配置信息

● 我在Django项目中添加了一个App, 名称为img_cos, 在该App中新建了一个config.py文件, 在中写入了关于腾讯云存储的相关配置信息。

appid 已在配置中移除,请在参数 Bucket 中带上 appid。Bucket 由 BucketName-APPID 组成 # 1. 设置用户配置, 包括 secretId, secretKey 以及 Region # -*- coding=utf-8 import logging import os import sys import uuid from PIL import Image from qcloud_cos import CosConfig, CosServiceError from qcloud_cos import CosS3Client from my_blog import settings

logging.basicConfig(level=logging.INFO, stream=sys.stdout)

secret_id = '******' # 替换为用户的 secretId secret_key = '*****' # 替换为用户的 SecretKey region = 'ap-shanghai' # 替换为用户的 Region token = None # 使用临时密钥需要传入 Token, 默认为空, 可不填 scheme = 'https' # 指定使用 http/https 协议来访问 COS, 默认为 https, 可不填 config = CosConfig(Region=region, SecretId=secret_id, SecretKey=secret_key, Token=token, cheme=scheme) # 2. 获取客户端对象 client = CosS3Client(config) host = 'https://zyk-1300089264.cos.ap-shanghai.myqcloud.com' # 存储桶访问地址 folder_path = '/article/' # 存储桶中保存图片文件夹名



上传图片

● 配置完毕之后,开始编写上传图片函数,在这里调用的SDK中的断点续传图片,支持断点续传,安性更高。

● 上传原理为:将上传的图片暂时保存至本地服务器,调用SDK中的上传图片方法(捕捉异常),若 传成功,则删除本地服务器上的图片,并返回图片在对象存储中的访问地址。

●由于图片名称可能会重复,因此采用UUID生成图片名称,保证名称唯一性。

为了节约空间,调用Pillow (没有安装Pillow库的小伙伴,可以先用 pip命令安装Pillow)中 thumbna l方法生成缩略图。具体代码如下。

```
# appid 已在配置中移除,请在参数 Bucket 中带上 appid。Bucket 由 BucketName-APPID 组成
# 1. 设置用户配置, 包括 secretId, secretKey 以及 Region
# -*- coding=utf-8
import logging
import os
import sys
import uuid
from PIL import Image
from gcloud cos import CosConfig, CosServiceError
from gcloud cos import CosS3Client
from my blog import settings
logging.basicConfig(level=logging.INFO, stream=sys.stdout)
secret id = '*****' # 替换为用户的 secretId
secret<sup>-</sup>key = '*****' # 替换为用户的 secretKey
region = 'ap-shanghai' # 替换为用户的 Region
token = None # 使用临时密钥需要传入 Token, 默认为空, 可不填
scheme = 'https' # 指定使用 http/https 协议来访问 COS, 默认为 https, 可不填
config = CosConfig(Region=region, SecretId=secret id, SecretKey=secret key, Token=token,
cheme=scheme)
# 2. 获取客户端对象
client = CosS3Client(config)
host = 'https://zyk-1300089264.cos.ap-shanghai.myqcloud.com' #存储桶访问地址
folder path = '/article/' #存储桶中保存图片文件夹名
# 上传图片
def upload file senior(file):
  file name = create file name(file) # 根据UUID生成文件名
  root path = os.path.join(settings.MEDIA ROOT, 'vditor') # 生成文件上传目录
  if not os.path.exists(root path): # 判断文件上传目录是否存在
    os.makedirs(root path) # 不存在,则创建
  file path = os.path.join(root path, file name)
  with open(file path, 'wb') as f: # 文件流将图片写入本地
    for c in file.chunks():
      f.write(c)
  file path = compress img(file path, 0.75) # 压缩图片, 生成缩略图
  try: # 捕捉异常
    response = client.upload file( # 根据文件大小自动选择简单上传或分块上传, 分块上传具备
点续传功能。
      Bucket='zyk-1300089264', #存储桶名称
      LocalFilePath=file path, # 本地图片路径
      Key=folder path + file name, # 上传路径
```

```
PartSize=1.
      MAXThread=10,
      EnableMD5=False
    )
    if response['ETag'] != "":
      os.remove(file path) # 上传成功, 删除本地图片
      return host + folder path + file_name
  except CosServiceError as e:
    print(e.get digest msg())
    return None
#利用UUID生成文件名,防止重名
def create file name(file):
  type name = file.name[file.name.index('.'):] # 获取文件后缀
  file name = '{}{'.format(uuid.uuid4(), type_name) # 生成文件名
  return file name
#压缩图片 (file path为图片路径, rate为压缩率,压缩率范围为0~1)
def compress img(file path, rate):
  image = Image.open(file path) # 获得图像
  width = int(image.width * rate) # 宽
  height = int(image.height * rate) # 高
  image.thumbnail((width, height), Image.ANTIALIAS) # 生成缩略图
  image.save(file path) #保存
  return file path
```

● 编写视图函数,在 img cos/views.py中加入图片上传函数 (在这里利用了

djangorestframework库,未安装的小伙伴可以通过pip install djangorestframework命令安装此)。先从请求中获取图片,然后调用之前封装好的SDK上传图片方法,根据上传成功与否,封装json 据返回至前端,上传成功则将图片访问URL同时返回至前端,代码如下。

from rest_framework.decorators import api_view from rest_framework.response import Response from img_cos.config import upload_file, upload_file_senior

```
"""
上传图片至腾讯云COS
"""
@api_view(['POST']) # 只允许POST请求
def upload_to_cos(request):
    file = request.FILES.get('smfile') # 获取上传图片
    url = upload_file_senior(file) # 调用上传图片SDK, 返回图片访问URL
    if url is None: # 上传失败
        return Response({"msg": "上传失败", "code": 0})
    return Response({"msg": "上传成功", "code": 1, "url": url}) # 上传成功
```

● 将视图函数注册到 img_cos/urls.py中,如下。

```
from django.urls import path
from img cos import views
```

```
urlpatterns = [
path('upload_to_cos', views.upload_to_cos, name='upload_to_cos'), # 上传图片至腾讯云COS
]
```

4. 前端交互

● 获取上传图片文件对象,利用append()方法将其封装成FormData,调用ajax上传至后台。在这里要注意ajax的编写,需要添加**不处理**上传数据的参数。

● 具体思路:选择要上传的图片,然后设置一个上传图片按钮,为该按钮添加点击事件,事件中调用 传图片函数,用户点击此按钮即可实现图片上传。

● 在我的博客项目中,文章表单字段较多,而文章数据表的图片字段只保存图片访问URL。因此,根 ajax上传图片返回的数据,若上传成功,将返回的图片访问URL写入一个隐藏的 <input>标签,然后 提交表单。具体实现代码如下:

```
//上传文章贴图
function uploadArticleImg() {
  if ($('#img').val() === "" || $('#img').val() == null) { // 判断文件是否为空
     layer.msg("请选择要上传的图片");
     return;
  let formData = new FormData();
  formData.append('smfile', $('#img')[0].files[0]); //将文件对象写入formData
  $.ajax({
     type: 'POST',
     url: '{% url 'upload to cos' %}',
     headers: {'X-CSRFToken': getCsrfToken()},
     data: formData,
     dataType: 'JSON',
     contentType: false, //不处理数据
     processData: false,
     cache: false,
     success: function (data) {
       if (data.code = = = 1) {
         layer.msg(data.msg);
          $("#link url").val(data.url); // 将图片访问URL写入隐藏的input标签中
       } else {
         layer.msg(data.msg);
       }
    },
     error: function (data) {
       console.log(data);
    }
  });
}
```

5. 结果测试

● 运行Django项目,测试图片上传功能,选择要上传的图片,点击上传图片按钮,观察结果。

选择文件 下载.jpg



● 进入腾讯云控制台, 查看存储桶中是否生成刚刚上传的图片。

 6680 	× 💩 評議稿-控制台 × 😫 djangorestframewo	(k_西度理):			
← → C a console.cld	oud.tencent.com/cos5/bucket/setting?type=filelist&bucketNam	e=zyk-13			
🌐 应用 🌖 Django 🜖 朝調	🤒 遊戏 🤗 視然 🧐 軟件透滞 🤚 VPN 🔮 百度 🧐 git	mysql			
🙆 🚓 🖘 🖬 •	PEMARK +				
対象存储	22295b75-114d-460e-b685079deceb.jpg	8.15KB			
12 RED.	271767a6-eaba-42cf-bbe5-o4d630e3c8b7.png	1.60MB			
C76618788	2c867b00-ded6-4d4b-8661-0bb83b996768.jpg	2.06MB			
C anena	2ct97e0b-83d5-429b-bfc5-88/3024045ca.png	134.75KE			
Series	2ecb01e8-d0b0-4a4d-bd5a-1ad902c5f796.pg	10.94KB			
© NEIFIE C	324dtt51-29a6-47b2-8478-ef52b990a41a.pg	10.94KB			
	37800fa4-7dt1-4262-b00f-a1cdtc901be0.png	160.96KE			
	3d617bb3-7d3a-4366-a215-54031c6a39dc.jpg	184.49KE			
	□ 3db/94al-b/09-40al-8396-d7a255e473e0 prg 🖌	38.80KB			
	3e6e427a-b3be-4f38-b015-729779bed256.pg	25.70K8			
	3/0/51/1-c/46-4148-90b9-2e97/5521fe9.jpg	10.94KB			
	41fcb150-ddd5-4875-b80a-5eec67e558e0.jpg				
	4588c845-a5e8-4351-82e8-3072db832d3f.jpg	90.79KB			
-	48da3b5c-20aa-4d36-b5e8-85478770c665 prg	52.33KB			

● 注意:如果想指定网站访问对象存储中的图片,需要在存储桶中的基础配置->跨域访问CORS设置 加自己网站的域名。 ← zyk-1300089264

Zyk	-1300089264					14	Change of the
文件列表	基础配置	域名管理	权限管理				
基本信息							
空间名称	zyk-1300089264						
所屋地城	上海 (中国) (ap-s)	hanghai)					
创建的间	2019-08-24 21:58:42						
访问城名	https://zyk-130008926	54.cos ap-shanghai	myqcloud.com 陷 (酒用于XML API)				
	相同地区的期间云内8 其他情况下使用,将8	6世界使用信域表対 動立 BGP 网境対 CO	COS 资源进行访问时,他收流最差。 S 资源进行访问,更多纳参考 默认成名访问指述	8 (Z			
	相同地区的确讯云内8 其他结论下使用,将1	6业务使用資源表対 MIZ BGP 開始対 CO	COS 资源进行访问时,免收流量者。 S 资源进行访问,更多纳参考 就认成名访问指动	• Ľ			
跨域访问	相同地区的侧吊云内的 其他情况下使用,将过 可CORS设置	61-54世紀高級おけ CO MIT BGP 開始対 CO	COS 资源进行访问时,免收流量素。 S 资源进行访问,更多请参考 就以成名访问指 解	8 12			
跨域访问 来클 Or	相同地区的制品类中的 其他结论下使用,将如 可CORS设置 rigin	6业务使用流域表对 基立 BGP 网络对 CO	COS 资源进行访问时,包收流量器。 S 资源进行访问,更多纳参考 就以越名访问和3 副作 Methods	Allow-Headers	Expose-Headers	超時 Max-Age	
跨域访问 来源 Or https://3	相同地区的明矾云内8 阿他県交下使用。特別 可 CORS设置 rigin 38 188 106 215 9 188 106 215	BLY CHIRALIN	COS 资源进行协同时,他收流基本。 S 按照进行访问,更多助参考 就以成名访问我的 展作 Methods PUT,GET,POST,DELETE,H EAD	Allow-Headers	Expose-Headers	部바 Max-Age 0	8

源加速回

Φ