# 水平仪识别系统

# 1、系统

## 1.1 ASGI config import os import django from django.core.asgi import get\_asgi\_application os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'LevelSystem.settings') # 配置 settings 模块 django.setup() from channels.routing import ProtocolTypeRouter, URLRouter from django.urls import path from LevelApp.consumers import DemoConsumer # application = get\_asgi\_application() django\_asgi\_application = get\_asgi\_application() appliction = ProtocolTypeRouter({ 'http': django\_asgi\_application, # 处理HTTP请求的Django应用程序 'websocket': URLRouter([ # 处理WebSocket连接的路由 path('ws/demo/', DemoConsumer.as\_asgi()), ]), })

## 1.2 Django settings

from pathlib import Path   
# Build paths inside the project like this: BASE\_DIR / 'subdir'.   
BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent   
# Quick-start development settings - unsuitable for production   
# See https://docs.djangoproject.com/en/4.2/howto/deployment/checklist/  
# SECURITY WARNING: keep the secret key used in production secret!   
SECRET\_KEY = 'django-insecure-cv%)\*q1u^f^er=3ct@9s%&#=\*utis\_o1df)=#y^fsp1x0j+&7i'   
# SECURITY WARNING: don't run with debug turned on in production!   
DEBUG = True   
ALLOWED\_HOSTS = []   
# settings.py   
"""   
百度API   
"""   
# top\_num: 返回的分类数量，不声明的话默认为 6 个   
BAIDU\_PARAMS = 6   
# 否则，留空 ACCESS\_TOKEN，于下方填入该模型部署的 API\_KEY 以及 SECRET\_KEY，会自动申请并显示新 ACCESS\_TOKEN   
BAIDU\_API\_KEY = ""   
BAIDU\_SECRET\_KEY = ""   
# 服务详情中的接口地址   
BAIDU\_MODEL\_API\_URL = "https://aip.baidubce.com/rpc/2.0/ai\_custom/v1/segmentation/shuipingyiV3"   
BAIDU\_TOKEN\_URL = "https://aip.baidubce.com/oauth/2.0/token?grant\_type=client\_credentials"   
"""   
minio 配置   
"""   
MINIO\_ACCESS\_KEY = 'minioadmin'   
MINIO\_SECRET\_KEY = 'minioadmin'   
MINIO\_ENDPOINT = '10.14.0.84:9000'   
MINIO\_BUCKET\_NAME = 'levelsystem' # 存储桶   
MINIO\_SECURE = False # Set to False if you are using HTTP instead of HTTPS   
DATA\_UPLOAD\_MAX\_MEMORY\_SIZE = 10485760 # 文件大小限制   
"""   
cap摄像机全局变量   
"""   
CAP = None   
CAP\_NUMBER = 0 # 摄像头，外接摄像头 1 为电脑摄像头   
"""   
跨域设置   
"""   
ALLOWED\_HOSTS = ['\*']   
"""   
jsonutil 文件配置   
"""   
JSON\_FILE\_PATH = "LevelApp/static/json/Atlas.json"   
JSON\_COUNT = "atlas"   
"""   
paddlepaddle 离线模型   
"""   
INTERFACE\_ADDRESS = 'http://10.13.169.245:12138/img\_prediction'   
"""   
误差计算   
"""   
DISPLACEMENTS = 'displacements'   
LEVEL\_LENGTH = 150 # 水平仪长度   
"""   
utils.py   
"""   
MEDIA\_ROOT = "./"   
# 定义全局变量 websocket\_channels   
WEBSOCKET\_CHANNELS = set()   
CORS\_ALLOW\_ALL\_ORIGINS = True   
# Application definition   
INSTALLED\_APPS = [   
 'django.contrib.admin',   
 'django.contrib.auth',   
 'django.contrib.contenttypes',   
 'django.contrib.sessions',   
 'django.contrib.messages',   
 'django.contrib.staticfiles',   
 'LevelApp.apps.LevelappConfig',   
 'corsheaders',   
]   
MIDDLEWARE = [   
 'django.middleware.security.SecurityMiddleware',   
 'django.contrib.sessions.middleware.SessionMiddleware',   
 'django.middleware.common.CommonMiddleware',   
 'django.middleware.csrf.CsrfViewMiddleware',   
 'django.contrib.auth.middleware.AuthenticationMiddleware',   
 'django.contrib.messages.middleware.MessageMiddleware',   
 'django.middleware.clickjacking.XFrameOptionsMiddleware',   
 'corsheaders.middleware.CorsMiddleware',   
 'django.middleware.common.CommonMiddleware',   
 'LevelApp.middleware.MyCustomExceptionMiddleware',   
]   
ROOT\_URLCONF = 'LevelSystem.urls'   
TEMPLATES = [   
 {   
 'BACKEND': 'django.template.backends.django.DjangoTemplates',   
 'DIRS': [],   
 'APP\_DIRS': True,   
 'OPTIONS': {   
 'context\_processors': [   
 'django.template.context\_processors.debug',   
 'django.template.context\_processors.request',   
 'django.contrib.auth.context\_processors.auth',   
 'django.contrib.messages.context\_processors.messages',   
 ],   
 },   
 },   
]   
WSGI\_APPLICATION = 'LevelSystem.wsgi.application'   
# Database   
# https://docs.djangoproject.com/en/4.2/ref/settings/#databases   
DATABASES = {   
 'default': {   
 'ENGINE': 'django.db.backends.sqlite3',   
 'NAME': BASE\_DIR / 'db.sqlite3',   
 }   
}   
# Password validation   
# https://docs.djangoproject.com/en/4.2/ref/settings/#auth-password-validators   
AUTH\_PASSWORD\_VALIDATORS = [   
 {   
 'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',   
 },   
 {   
 'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',   
 },   
 {   
 'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',   
 },   
 {   
 'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',   
 },   
]   
# Internationalization   
# https://docs.djangoproject.com/en/4.2/topics/i18n/   
LANGUAGE\_CODE = 'en-us'   
TIME\_ZONE = 'UTC'   
USE\_I18N = True   
USE\_TZ = True   
# Static files (CSS, JavaScript, Images)   
# https://docs.djangoproject.com/en/4.2/howto/static-files/   
STATIC\_URL = 'static/'   
# Default primary key field type   
# https://docs.djangoproject.com/en/4.2/ref/settings/#default-auto-field   
DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

## 1.3 URL configuration

"""   
URL configuration for LevelSystem project.   
The `urlpatterns` list routes URLs to views. For more information please see:   
 https://docs.djangoproject.com/en/4.2/topics/http/urls/   
Examples:   
Function views   
 1. Add an import: from my\_app import views   
 2. Add a URL to urlpatterns: path('', views.home, name='home')   
Class-based views   
 1. Add an import: from other\_app.views import Home   
 2. Add a URL to urlpatterns: path('', Home.as\_view(), name='home')   
Including another URLconf   
 1. Import the include() function: from django.urls import include, path   
 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))   
"""   
from django.contrib import admin   
from django.urls import path   
from LevelApp import views   
urlpatterns = [   
 # path('admin/', admin.site.urls),   
 # 表示用户一访问此接口 就会去到我们注册的app中的 views文件中寻找index函数并执行   
 path('html/', views.html),   
 path('baidu/', views.baidu),   
 path('upload\_photo/', views.upload\_photo),   
 path('photograph/', views.photograph, name='photograph'),   
 path('openCamera/', views.openCamera, name='openCamera'),   
 path('closeCamera/', views.closeCamera, name='closeCamera'),   
 path('get\_list/', views.get\_list, name='get\_list'),   
 path('save/', views.save, name='save'),   
 path('update/', views.update, name='update'),   
 path('delete/', views.delete, name='delete'),   
 path('pp\_upload/', views.pp\_upload, name='pp\_upload'),   
 path('statistical\_errors/', views.statistical\_errors, name='statistical\_errors'),   
 path('pp\_photograph/', views.pp\_photograph, name='pp\_photograph'),   
]

## 1.4 WSGI config

"""   
WSGI config for LevelSystem project.   
It exposes the WSGI callable as a module-level variable named ``application``.   
For more information on this file, see   
https://docs.djangoproject.com/en/4.2/howto/deployment/wsgi/   
"""   
import os   
from dango.core.wsgi import get\_wsgi\_application   
os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'LevelSystem.settings')  
application = get\_wsgi\_application()

## 1.5 main

#!/usr/bin/env python   
"""Django's command-line utility for administrative tasks."""   
import os   
import sys   
def main():   
 """Run administrative tasks."""   
 os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'LevelSystem.settings')   
 try:   
 from django.core.management import execute\_from\_command\_line   
 except ImportError as exc:   
 raise ImportError(   
 "Couldn't import Django. Are you sure it's installed and "   
 "available on your PYTHONPATH environment variable? Did you "   
 "forget to activate a virtual environment?"   
 ) from exc   
 execute\_from\_command\_line(sys.argv)   
if \_\_name\_\_ == '\_\_main\_\_':   
 main()

# 2、 工具类

## 2.1 JSON UTIL

import json   
from django.conf import settings   
# 读取JSON文件   
def read\_json\_file(file\_path):   
 with open(file\_path, 'r') as file:   
 data = json.load(file)   
 return data   
# 写入JSON文件   
def write\_json\_file(file\_path, data):   
 with open(file\_path, 'w') as file:   
 json.dump(data, file, indent=2)   
# 增加数据   
def add\_data\_json(data, atlas\_type, new\_data):   
 atlas = data.get(atlas\_type, [])   
 atlas.append(new\_data)   
 data[atlas\_type] = atlas   
 # 写入JSON文件   
 write\_json\_file(settings.JSON\_FILE\_PATH, data)   
 return True   
# 删除数据   
def delete\_data(data, atlas\_type, id\_to\_delete):   
 atlas = data.get(atlas\_type, [])   
 data[atlas\_type] = [item for item in atlas if item['id'] != int(id\_to\_delete)]   
 print(data[atlas\_type])   
 # 写入JSON文件   
 write\_json\_file(settings.JSON\_FILE\_PATH, data)   
 return True   
# 修改数据   
def update\_data(data, atlas\_type, updated\_data):   
 atlas = data.get(atlas\_type, [])   
 for item in atlas:   
 if item['id'] == updated\_data['id']:   
 item.update(updated\_data)   
 # 写入JSON文件   
 write\_json\_file(settings.JSON\_FILE\_PATH, data)   
 return True   
 return False   
# 查询数据   
def query\_data(data, atlas\_type, id\_to\_query):   
 atlas = data.get(atlas\_type, [])   
 for item in atlas:   
 if item['id'] == id\_to\_query:   
 return item   
 return None   
# 查询所有atlas\_type 类型中的所有数据   
def query\_all\_issue\_data(data, atlas\_type):   
 return data.get(atlas\_type, [])

## 2.2 误差计算

from loguru import logger   
   
   
def calculate\_straightness\_error(an\_values, B):   
 n = len(an\_values)   
 # 计算折线纵坐标总值   
 total\_sum = sum(an\_values)   
 # 计算首尾线各点的纵坐标值   
 line\_values = [total\_sum \* (i / n) for i in range(1, n + 1)]   
 # 初始化折线纵坐标值列表   
 broken\_line\_values = []   
 # 计算折线纵坐标值   
 for i in range(n):   
 if i == 0:   
 broken\_line\_values.append(an\_values[i])   
 else:   
 broken\_line\_values.append(an\_values[i] + broken\_line\_values[i - 1])   
 # 计算折线各点的纵坐标值与首尾线各点的纵坐标差值   
 delta\_values = [broken\_line\_values[i] - line\_values[i] for i in range(n)]   
 # 查询纵坐标差值最大值、最小值   
 max\_delta = max(delta\_values)   
 min\_delta = min(delta\_values)   
 # 直线度误差计算值   
 S = abs(max\_delta) + abs(min\_delta)   
 S = S \* 0.02 / 1000 \* B   
 logger.info(f'本次计算的偏移数组为:{an\_values},水平仪的长度为:{B},所计算的误差为:{S}')   
 return S

## 2.3 MINIO UTIL

from django.conf import settings   
from django.core.files.storage import default\_storage   
from django.http import JsonResponse   
from django.views.decorators.csrf import csrf\_exempt   
from minio import Minio   
from urllib3.exceptions import ResponseError   
import datetime   
"""   
上传图片   
"""   
def upload\_photo(photo):   
 try:   
 minio\_client = Minio(   
 settings.MINIO\_ENDPOINT,   
 access\_key=settings.MINIO\_ACCESS\_KEY,   
 secret\_key=settings.MINIO\_SECRET\_KEY,   
 secure=settings.MINIO\_SECURE,   
 )   
 # Ensure the bucket exists   
 bucket\_name = settings.MINIO\_BUCKET\_NAME   
 if not minio\_client.bucket\_exists(bucket\_name):   
 minio\_client.make\_bucket(bucket\_name)   
 # Generate a unique file name   
 current\_datetime = datetime.datetime.now().strftime('%Y%m%d%H%M%S')   
 file\_name = f"{current\_datetime}/{photo.name}"   
 # Upload the photo to MinIO   
 minio\_client.put\_object(   
 bucket\_name,   
 file\_name,   
 photo,   
 length=photo.size,   
 content\_type=photo.content\_type,   
 )   
 # Generate a pre-signed URL for the uploaded photo   
 photo\_url = minio\_client.presigned\_get\_object(bucket\_name, file\_name)   
 return photo\_url   
 except Exception as e:   
 print("Error:", str(e))   
 return JsonResponse({'error': 'Internal Server Error'}, status=500)   
def upload\_photo\_stream(photo, image\_name):   
 try:   
 minio\_client = Minio(   
 settings.MINIO\_ENDPOINT,   
 access\_key=settings.MINIO\_ACCESS\_KEY,   
 secret\_key=settings.MINIO\_SECRET\_KEY,   
 secure=settings.MINIO\_SECURE,   
 )   
 # Ensure the bucket exists   
 bucket\_name = settings.MINIO\_BUCKET\_NAME   
 if not minio\_client.bucket\_exists(bucket\_name):   
 minio\_client.make\_bucket(bucket\_name)   
 # Generate a unique file name   
 current\_datetime = datetime.datetime.now().strftime('%Y%m%d')   
 file\_name = f"{current\_datetime}/{image\_name}"   
 # Create a new BytesIO object and copy the contents of image\_byte\_stream into it   
 # photo\_copy = io.BytesIO()   
 photo.seek(0) # Move the pointer to the beginning of the stream   
 # photo\_copy.write(photo.read()) # Copy the data from image\_byte\_stream to photo\_copy   
 # Upload the photo to MinIO   
 minio\_client.put\_object(   
 bucket\_name,   
 file\_name,   
 photo,   
 length=photo.getbuffer().nbytes,   
 content\_type='image/png',   
 )   
 # Generate a pre-signed URL for the uploaded photo   
 photo\_url = minio\_client.presigned\_get\_object(bucket\_name, file\_name)   
 return photo\_url   
 except Exception as e:   
 print("Error:", str(e))   
 return JsonResponse({'error': 'Internal Server Error'}, status=500)

## 2.4 拍照

# -\*- coding:utf-8 -\*-   
"""   
描述作用：使用OpenCv提供的函数，进行摄像头拍照。按q键进行单张拍照，按t键拍照退出   
使用说明: 直接运行即可，若程序报错，请检查打开摄像头索引方式   
硬件信息：   
硬件描述：工业级高清单目摄像头，H043-8814-V2。USB免驱动   
购买链接："https://m.tb.cn/h.5YiNvc1?tk=glFTdGEci8S CZ3457"   
"""   
import time   
import cv2   
import io   
from loguru import logger   
from django.conf import settings   
from LevelApp.utils import minioUtils   
from LevelApp.utils import utils   
def getCameraQuantity() -> int:   
 """获取可用摄像头数量"""   
 quantity = 0   
 cap = cv2.VideoCapture() # 视频流对象   
 index = 0   
 # 一般一台电脑连接的摄像头数量不会超过5个   
 # 普遍情况下最多两个   
 while index < 5:   
 ret = cap.open(index)   
 if ret:   
 quantity += 1 # 可用摄像头数量+1   
 cap.release() # 释放打开的摄像头   
 index += 1 # 索引+1   
 else:   
 break # 一旦遇到打开失败的序号，则没有更多的摄像头了   
 return quantity   
def getTimeStampForImage() -> str:   
 """   
 获取当前时间，格式：年月日   
 :return:   
 """   
 from datetime import datetime   
 # 获取当前时间戳   
 timestamp = datetime.timestamp(datetime.now())   
 dt\_object = datetime.fromtimestamp(timestamp)   
 # 格式化成字符串   
 formatted\_date = dt\_object.strftime("%Y%m%d%H%M%S")   
 # 打印结果   
 return formatted\_date   
def getPicturePATH():   
 # 设置文件名   
 currentTimestamp = getTimeStampForImage()   
 fileName = f"C{currentTimestamp}.jpg" # 组合时间戳图片名   
 logger.info("拍摄文件名:" + fileName)   
 pictureAddress = f"./images/{fileName}" # 文件相对保存路径   
 return pictureAddress, fileName   
def timing(func):   
 """   
 计算程序运行耗时   
 :param func: 函数，修饰器用法   
 :return: 内部函数   
 """   
 def inner():   
 startTime = time.time()   
 print("开始时间：", startTime)   
 func()   
 endTime = time.time()   
 print("结束时间：", endTime)   
 timeDiff = endTime - startTime # 计算的时间差为程序的执行时间，单位为秒/s   
 logger.info("本次拍摄耗时 时间间隔（单位/秒):" + str(timeDiff))   
 return inner  
def readVideo(boxHeight, boxWidth) -> str:   
 """ 拍照 """   
def openCamera(number):   
 """   
 打开摄像头   
 :param number: 摄像头编号 0表示本机电脑摄像头   
 :return: 摄像头状态,相应摄像头信息   
 """   
 # 打开摄像头   
 logger.info("摄像头正在打开中……")   
 cap = cv2.VideoCapture(number)   
 if cap.isOpened():   
 logger.info("摄像头已打开~")   
 return True, cap   
 logger.error("摄像头打开失败，请检查摄像头是否连接正常！！")   
 cap= None   
 return False, cap   
def takePhoto(cap):   
 """   
 拍照   
 :param cap: 摄像头信息   
 :return:   
 """   
 try:   
 # 读取摄像头画面，ret为是否成功打开摄像头,frame为视频的每一帧图像   
 ret, frame = cap.read()   
 if not ret:   
 logger.error("无法获取摄像头画面,请检查摄像头状态!")   
 return False, None   
 # 获取拍照相对路径   
 pictureAddress, fileName = getPicturePATH()   
 # 保存校准框内的图像   
 cv2.imwrite(pictureAddress, frame) # 图片保存路径   
 # 保存校准框内的图像   
 image\_byte\_stream = io.BytesIO()   
 # 将图像转换为字节流   
 ret, buffer = cv2.imencode('.jpg', frame)   
 if ret:   
 image\_byte\_stream.write(buffer)   
 image\_byte\_stream.seek(0)   
 results = utils.picToBase64AndRequestBaiDuApiV2(None, image\_byte\_stream.read(), fileName)   
 print(results)   
 logger.info("拍照成功~")   
 return True, results   
 return False, None   
 except Exception as e:   
 logger.error(e)   
 return False, None   
"""   
paddlepaddle 离线模型调用方法   
"""   
def take\_photo\_pp(cap):   
 """   
 拍照   
 :param cap: 摄像头信息   
 :return:   
 """   
 try:   
 # 读取摄像头画面，ret为是否成功打开摄像头,frame为视频的每一帧图像   
 ret, frame = cap.read()   
 if not ret:   
 logger.error("无法获取摄像头画面,请检查摄像头状态!")   
 return False, None   
 # 获取拍照相对路径   
 pictureAddress, fileName = getPicturePATH()   
 # 保存校准框内的图像   
 cv2.imwrite(pictureAddress, frame) # 图片保存路径   
 # 保存校准框内的图像   
 image\_byte\_stream = io.BytesIO()   
 # 将图像转换为字节流   
 ret, buffer = cv2.imencode('.jpg', frame)   
 if ret:   
 image\_byte\_stream.write(buffer)   
 image\_byte\_stream.seek(0)   
 logger.info("拍照成功~")   
 return True, image\_byte\_stream.read(), fileName   
 return False, None, None   
 except Exception as e:   
 logger.error(e)   
 return False, None, None   
def closeCamera(cap):   
 """   
 关闭摄像头   
 :param cap: 摄像头信息   
 """   
 # 释放摄像头资源   
 if cap.isOpened():   
 cap.release()   
 logger.info("摄像头已成功关闭~")   
 return True   
 logger.error("无法关闭摄像头，该摄像头已关闭！")   
def main():   
 result, cap = openCamera(1) # 0代表本机摄像头   
 if result:   
 settings.CAP = cap   
 result, results = takePhoto(settings.CAP)   
 result = closeCamera(settings.CAP)   
if \_\_name\_\_ == '\_\_main\_\_':   
 main()

## 2.5 飞浆AI

import matplotlib   
   
# 作为非交互的模式避免线程冲突   
matplotlib.use('Agg')   
import matplotlib.pyplot as plt   
from PIL import Image   
import requests   
import io   
import json   
from loguru import logger   
import base64   
import math   
from datetime import datetime   
import os   
from django.conf import settings   
from LevelApp.utils import minioUtils   
from django.http import JsonResponse   
from LevelApp.utils import jsonUtil   
# 生成当前时间戳，用于拼接日志文件名   
timestamp = datetime.now().strftime("%Y-%m-%d")   
logger.add(f"logs/baiduApi\_{timestamp}.log", retention="1 days", level='INFO')   
   
# 设置支持中文字符的字体为 "SimHei"   
plt.rcParams['font.family'] = 'SimHei'  
def picToBase64AndRequestBaiDuApiV2(ACCESS\_TOKEN=None, image\_data=None, image\_name=None):   
 # top\_num: 返回的分类数量，不声明的话默认为 6 个   
 PARAMS = {"top\_num": settings.BAIDU\_PARAMS}   
 # 否则，留空 ACCESS\_TOKEN，于下方填入该模型部署的 API\_KEY 以及 SECRET\_KEY，会自动申请并显示新 ACCESS\_TOKEN   
 API\_KEY = settings.BAIDU\_API\_KEY   
 SECRET\_KEY = settings.BAIDU\_SECRET\_KEY   
 # 服务详情中的接口地址   
 DEL\_API\_URL = settings.BAIDU\_MODEL\_API\_URL   
 base64\_data = base64.b64encode(image\_data)   
 base64\_str = base64\_data.decode('UTF8')   
 PARAMS["image"] = base64\_str   
 if not ACCESS\_TOKEN:   
 auth\_url = settings.BAIDU\_TOKEN\_URL + "&client\_id={}&client\_secret={}".format(   
 API\_KEY, SECRET\_KEY)   
 auth\_resp = requests.get(auth\_url)   
 auth\_resp\_json = auth\_resp.json()   
 ACCESS\_TOKEN = auth\_resp\_json["access\_token"]   
 else:   
 logger.info("新 ACCESS\_TOKEN: {}".format(ACCESS\_TOKEN))   
 request\_url = "{}?access\_token={}".format(MODEL\_API\_URL, ACCESS\_TOKEN)   
 response = requests.post(url=request\_url, json=PARAMS)   
 response\_json = response.json()   
 response\_str = json.dumps(response\_json, indent=4, ensure\_ascii=False)   
 logger.info("结果:{}".format(response\_str))   
 results = response\_json["results"]   
 photo\_url = draw\_all\_shapes(image\_data, results, image\_name)   
 return photo\_url   
def draw\_all\_shapes(image\_data, results, image\_name):   
 try:   
 # 将图片数据转换为PIL图像   
 img = Image.open(io.BytesIO(image\_data))   
 # img = Image.open(image\_path)   
 plt.imshow(img)   
 plt.axis('off') # 关闭坐标轴标签   
 ax = plt.gca()   
 colors = ['r', 'g', 'b', 'c', 'm', 'y']  
 # 绘制所有识别到的图形   
 for i, result in enumerate(results):   
 location = result["location"]   
 color = colors[i % len(colors)]   
 left, top, right, bottom = (   
 location["left"],   
 location["top"],   
 location["left"] + location["width"],   
 location["top"] + location["height"],   
 )   
 rect = plt.Rectangle((left, top), location["width"], location["height"],   
 linewidth=1, edgecolor=color, facecolor='none')   
 ax.add\_patch(rect)   
 # 根据 top 值对刻度点进行排序   
 results\_sorted = sorted([result for result in results if result["name"] == "刻度点"],   
 key=lambda x: x["location"]["top"])   
 # 寻找 top 值最相近的两个刻度点   
 min\_diff = float('inf')   
 closest\_ticks = None   
 for i in range(len(results\_sorted) - 1):   
 diff = abs(results\_sorted[i]["location"]["top"] - results\_sorted[i + 1]["location"]["top"])   
 if diff < min\_diff:   
 min\_diff = diff   
 closest\_ticks = [results\_sorted[i], results\_sorted[i + 1]]   
 """   
 计算出两根刻度线之间的距离   
 """   
 ScaleLineDistance = abs(closest\_ticks[0]["location"]["left"] - closest\_ticks[1]["location"]["left"]) / 6   
 logger.info("两根刻度线之间的距离：'{}'".format(ScaleLineDistance))   
 center\_point = {}   
 # 绘制两个刻度点的中心点   
 for i, tick in enumerate(closest\_ticks):   
 location = tick["location"]   
 color = colors[i % len(colors)]   
 left, top, right, bottom = (   
 location["left"],   
 location["top"],   
 location["left"] + location["width"],   
 location["top"] + location["height"],   
 )   
 center\_x = (left + right) / 2   
 center\_y = (top + bottom) / 2   
 plt.scatter(center\_x, center\_y, color=color, marker='x', s=10, label=f'刻度点 {i + 1}')   
 center\_point["point" + str(i)] = [center\_x, center\_y]   
 # 输出 top 值   
 center\_x1, center\_y1 = center\_point["point0"]   
 center\_x2, center\_y2 = center\_point["point1"]   
 # 绘制连接线   
 # 绘制连接线   
 plt.plot([center\_x1, center\_x2], [center\_y1, center\_y2], color='k', linestyle='-', linewidth=1, label='top连接线')   
 # 计算连接线的中心点坐标   
 center\_x\_line = (center\_x1 + center\_x2) / 2   
 center\_y\_line = (center\_y1 + center\_y2) / 2   
 # 绘制连接线的中心点   
 plt.scatter(center\_x\_line, center\_y\_line, color='y', marker='o', s=10, label='top连接线中心点')   
 top\_center\_point = [center\_x\_line, center\_y\_line] # 保存中心点   
 # 根据 left 值对刻度点进行排序   
 results\_sorted\_left = sorted([result for result in results if result["name"] == "刻度点"],   
 key=lambda x: x["location"]["left"])   
 # 寻找 top 值最相近的两个刻度点   
 min\_diff\_left = float('inf')   
 closest\_ticks\_left = None   
 print(min\_diff\_left)   
 for i in range(len(results\_sorted\_left) - 1):   
 diff = abs(results\_sorted\_left[i]["location"]["left"] - results\_sorted\_left[i + 1]["location"]["left"])   
 if diff < min\_diff\_left:   
 min\_diff\_left = diff   
 closest\_ticks\_left = [results\_sorted\_left[i], results\_sorted\_left[i + 1]]   
 # 绘制两个刻度点的中心点   
 for i, tick in enumerate(closest\_ticks\_left):   
 location = tick["location"]   
 color = colors[i % len(colors)]   
 left, top, right, bottom = (   
 location["left"],   
 location["top"],   
 location["left"] + location["width"],   
 location["top"] + location["height"],   
 )   
 center\_x = (left + right) / 2   
 center\_y = (top + bottom) / 2   
 plt.scatter(center\_x, center\_y, color=color, marker='x', s=10, label=f'刻度点 {i + 1}')   
 center\_point["point" + str(i)] = [center\_x, center\_y]   
 # 输出 top 值   
 # plt.text(center\_x, center\_y, str(location["top"]), fontsize=12, color=color, ha='center', va='bottom')   
 center\_x1, center\_y1 = center\_point["point0"]   
 center\_x2, center\_y2 = center\_point["point1"]   
 # 绘制连接线   
 plt.plot([center\_x1, center\_x2], [center\_y1, center\_y2], color='k', linestyle='-', linewidth=1, label='left连接线')   
 # 计算连接线的中心点坐标   
 center\_x\_line = (center\_x1 + center\_x2) / 2   
 center\_y\_line = (center\_y1 + center\_y2) / 2   
 # 绘制连接线的中心点   
 plt.scatter(center\_x\_line, center\_y\_line, color='b', marker='o', s=10, label='left 连接线中心点')   
 left\_center\_point = [center\_x\_line, center\_y\_line] # 保存中心点  
 """   
 绘制刻度线的中心点   
 """   
 tickMarks\_center\_point = [top\_center\_point[0], left\_center\_point[1]]   
 plt.scatter(tickMarks\_center\_point[0], tickMarks\_center\_point[1], color='g', marker='o', s=10, label='刻度线中心点')   
 logger.info("度线中心点：'{}'".format(tickMarks\_center\_point))   
 bubles\_center\_point = []   
 bubbles\_distances = []   
 # 绘制气泡的中心点   
 bubbles = [result for result in results if result["name"] == "气泡"]   
 for i, bubble in enumerate(bubbles):   
 location = bubble["location"]   
 color = colors[i % len(colors)]   
 left, top, right, bottom = (   
 location["left"],   
 location["top"],   
 location["left"] + location["width"],   
 location["top"] + location["height"],   
 )   
 center\_x = (left + right) / 2   
 center\_y = (top + bottom) / 2   
 plt.scatter(center\_x, center\_y, color=color, marker='o', s=10, label='气泡中心点')   
 bubbles\_center\_point = [center\_x, center\_y] # 保存气泡中心点   
 # 计算气泡轮廓左上角和右上角的点之间的距离（水平距离）   
 distance = right - left   
 bubbles\_distances.append(distance) # 保存气泡轮廓的距离  
 """   
 计算气泡轮廓左侧到右侧的距离的二分之一 ScaleLineDistance   
 """   
 bubbles\_distance\_half = bubbles\_distances[0] / 2   
 logger.info("气泡轮廓左侧到右侧的距离的二分之一：'{}'".format(bubbles\_distance\_half))   
 """   
 绘制连接线 并计算距离   
 """   
 logger.info("气泡中心点：'{}'".format(bubbles\_center\_point))   
 # 计算两点之间的距离   
 distance = distance\_between\_points(tickMarks\_center\_point[0], tickMarks\_center\_point[1],   
 bubbles\_center\_point[0], bubbles\_center\_point[1])   
 logger.info("两点之间的距离：'{}'".format(distance))   
 bubble\_left\_scale = (distance - bubbles\_distance\_half) / ScaleLineDistance   
 bubble\_right\_scale = (distance + bubbles\_distance\_half) / ScaleLineDistance   
 logger.info("气泡在刻度线：'第{}根'到'{}根之间'".format(bubble\_left\_scale, bubble\_right\_scale))   
 # 添图例   
 plt.legend(loc='upper right', fontsize=7)   
 # 计算偏移   
 displacement = distance / ScaleLineDistance \* 30   
 """   
 计算气泡在刻度线中心点的左侧还是右侧   
 """   
 bubble\_placement = ''   
 if (tickMarks\_center\_point[0] - bubbles\_center\_point[0]) > 0:   
 bubble\_placement = '左侧'   
 displacement = -displacement   
 elif (tickMarks\_center\_point[0] - bubbles\_center\_point[0]) == 0  
 bubble\_placement = '中心'   
 else:   
 bubble\_placement = '右侧'   
 logger.info("气泡在刻度线中心点的：'{}'".format(bubble\_placement)  
 logger.info("气泡偏移为：'{}'".format(displacement))   
 # 保存绘制的图片到本地   
 # save\_path = os.path.join(settings.MEDIA\_ROOT, "result\_image.png")   
 # plt.savefig(save\_path, bbox\_inches='tight')   
 # plt.show()   
 # 将绘制的图像保存到字节流中而不是本地文件   
 image\_byte\_stream = io.BytesIO()   
 plt.savefig(image\_byte\_stream, format='png', bbox\_inches='tight')   
 上传到minio中   
 photo\_url = minioUtils.upload\_photo\_stream(image\_byte\_stream, image\_name)   
 plt.close()   
 # 读取json文件   
 json\_data = jsonUtil.read\_json\_file(settings.JSON\_FILE\_PATH)   
 # 写如json中   
 jsonUtil.add\_data\_json(json\_data, settings.DISPLACEMENTS, displacement)   
 photo\_results = {}   
 photo\_results.setdefault('ScaleLineDistance', ScaleLineDistanc  
 photo\_results.setdefault('tickMarks\_center\_point', tickMarks\_center\_point)   
 photo\_results.setdefault('bubble\_placement', bubble\_placement)   
 photo\_results.setdefault('bubbles\_distance\_half', bubbles\_distance\_half)   
 photo\_results.setdefault('bubbles\_center\_point', bubbles\_center\_point)   
 photo\_results.setdefault('distance', distance)   
 photo\_results.setdefault('bubble\_left\_scale', bubble\_left\_scale  
 photo\_results.setdefault('bubble\_right\_scale', bubble\_right\_scale)  
 photo\_results.setdefault('photo\_url', photo\_url)   
 photo\_results.setdefault('displacements', jsonUtil.query\_all\_issue\_data(json\_data, settings.DISPLACEMENTS))   
 return photo\_results   
 except Exception as e:   
 print("Error:", str(e))   
 return JsonResponse({'error': f'绘图时出现错误:{e}', 'code': 500}, status=500)   
def distance\_between\_points(x1, y1, x2, y2):   
 return math.sqrt((x2 - x1) \*\* 2 + (y2 - y1) \*\* 2)   
"""   
电脑像素值   
def ppi():   
 # Given values   
 horizontal\_pixels = 2560   
 vertical\_pixels = 1440   
 screen\_size\_inch = 15.6   
 # Calculate PPI   
 ppi = math.sqrt(horizontal\_pixels \*\* 2 + vertical\_pixels \*\* 2) / screen\_size\_inch   
 return ppi

# 3、APP

## 3.1 APP

from django.apps import AppConfig   
   
class LevelappConfig(AppConfig):   
 default\_auto\_field = 'django.db.models.BigAutoField'   
 name = 'LevelApp'

## 3.2 中间件

# myapp/middleware.py   
from django.http import JsonResponse   
from loguru import logger   
class MyCustomExceptionMiddleware:   
 def \_\_init\_\_(self, get\_response):   
 self.get\_response = get\_response   
 def \_\_call\_\_(self, request):   
 response = self.get\_response(request)   
 return response   
 def process\_exception(self, request, exception):   
 error\_message = str(exception) # 将异常信息转换为字符  
 logger.info(f'出现全局异常{error\_message}')   
 return JsonResponse({'data': f'出现全局异常{error\_message}', 'code': 500}, status=500)

## 3.3 view

from django.conf import settings   
from django.shortcuts import render, HttpResponse   
from django.views.decorators.csrf import csrf\_exempt   
# Create your views here.   
from LevelApp.utils import utils   
from LevelApp.utils import minioUtils   
from LevelApp.utils import TakePhotoForAIUtil   
from LevelApp.utils import jsonUtil   
from LevelApp.utils import linear\_straightness\_checker   
import json   
import requests   
import base64   
import io   
"""   
pip install django-storages   
pip install channels   
pip install django-cors-headers   
"""   
from channels.layers import get\_channel\_layer   
from asgiref.sync import async\_to\_sync   
# from django.core.files.storage import default\_storage   
from django.http import JsonResponse   
# from django.views.decorators.csrf import csrf\_exempt   
# from minio import Minio   
# from urllib3.exceptions import ResponseError   
# import datetime   
@csrf\_exempt   
def upload\_photo(request):   
 try:   
 if request.method == 'POST' and request.FILES.get('photo'):   
 photo = request.FILES['photo']   
 url = minioUtils.upload\_photo(photo)   
 return JsonResponse({'imageUrl': url}, status=200)   
 else:   
 return JsonResponse({'error': 'Invalid request'}, status=400)   
 except Exception as e:   
 print("Error:", str(e))   
 return JsonResponse({'error': 'Internal Server Error'}, status=500)   
"""   
主页   
"""   
def html(request):   
 return render(request, "html.html")   
"""   
上传图片并识别   
"""   
@csrf\_exempt   
def baidu(request):   
 if request.method == 'POST' and request.FILES.get('image'):   
 uploaded\_image = request.FILES['image']   
 image\_name = uploaded\_image.name   
 results = utils.picToBase64AndRequestBaiDuApiV2(None, uploaded\_image.read(), image\_name)   
 results1 = []   
 results1.append(results)   
 return JsonResponse({'data': results1, 'code': 200}, status=20   
 return JsonResponse({'data': 'fail', 'code': 500}, status=500)   
"""   
调用摄像头拍照上传并识别   
"""   
@csrf\_exempt   
def photograph(request):   
 result, results = TakePhotoForAIUtil.takePhoto(settings.CAP)   
 if result:   
 results1 = []   
 results1.append(results)   
 return JsonResponse({'data': results1, 'code': 200}, status=200)  
 return JsonResponse({'data': 'fail', 'code': 500}, status=500)   
"""   
开启摄像头   
"""   
@csrf\_exempt   
def openCamera(request):   
 result, cap = TakePhotoForAIUtil.openCamera(settings.CAP\_NUMBER)   
 if result:   
 settings.CAP = cap   
 print(settings.CAP)   
 return JsonResponse({'data': 'success', 'code': 200}, status=200)   
 return JsonResponse({'data': 'fail', 'code': 500}, status=500)   
"""   
关闭摄像头   
"""   
@csrf\_exempt   
def closeCamera(request):   
 print(settings.CAP)   
 result = TakePhotoForAIUtil.closeCamera(settings.CAP)   
 if result:   
 return JsonResponse({'data': 'success', 'code': 200}, status=200)   
 return JsonResponse({'data': 'fail', 'code': 500}, status=500)   
查询照片   
"""   
@csrf\_exempt   
def get\_list(request):   
 atlas\_type = request.GET.get('atlas\_type', None)   
 data = jsonUtil.read\_json\_file(settings.JSON\_FILE\_PATH)   
 result = jsonUtil.query\_all\_issue\_data(data, atlas\_type)   
 if result:   
 return JsonResponse({'data': result, 'code': 200}, status=200)   
 return JsonResponse({'data': [], 'code': 200}, status=200)   
"""   
保存照片   
"""   
@csrf\_exempt   
def save(request):   
 if request.method == 'POST':   
 try:   
 json\_data = jsonUtil.read\_json\_file(settings.JSON\_FILE\_PAT  
 # 解析请求的JSON数据   
 data = json.loads(request.body)   
 atlas\_type = data.get('atlas\_type')   
 id = json\_data[settings.JSON\_COUNT][atlas\_type]   
 data["id"] = id # 设置id   
 json\_data[settings.JSON\_COUNT][atlas\_type] = id + 1 # 自  
 data\_json = jsonUtil.add\_data\_json(json\_data, atlas\_type, data)   
 return JsonResponse({'data': 'success', 'code': 200}, status=200)   
 except json.JSONDecodeError:   
 return JsonResponse({'data': 'fail', 'code': 500}, status=500)   
 else:   
 return JsonResponse({"error": "Only POST requests are allowed"}, status=500)   
@csrf\_exempt   
def update(request):   
 if request.method == 'POST':   
 try:   
 json\_data = jsonUtil.read\_json\_file(settings.JSON\_FILE\_PATH  
 # 解析请求的JSON数据   
 data = json.loads(request.body)   
 atlas\_type = data.get('atlas\_type')   
 data\_json = jsonUtil.update\_data(json\_data, atlas\_type, data)   
 return JsonResponse({'data': 'success', 'code': 200}, status=200)   
 except json.JSONDecodeError:   
 return JsonResponse({'data': 'fail', 'code': 500}, status=500)   
 else:   
 return JsonResponse({"error": "Only POST requests are allowed"}, status=500)   
"""   
删除   
"""   
@csrf\_exempt   
def delete(request):   
 atlas\_type = request.GET.get('atlas\_type', None)   
 id = request.GET.get('id', None)   
 print(atlas\_type)   
 print(id)   
 data = jsonUtil.read\_json\_file(settings.JSON\_FILE\_PATH)   
 result = jsonUtil.delete\_data(data, atlas\_type, id)   
 if result:   
 return JsonResponse({'data': result, 'code': 200}, status=200)   
 return JsonResponse({'data': 'fail', 'code': 500}, status=500)   
"""   
paddlepaddle 离线模型   
"""   
@csrf\_exempt   
def pp\_upload(request):   
 if request.method == 'POST' and request.FILES.get('image'):   
 uploaded\_image = request.FILES['image']   
 image\_name = uploaded\_image.name   
 files = {'image': uploaded\_image.read()}   
 # 请求模型   
 response = requests.post(settings.INTERFACE\_ADDRESS, files=files)   
 loads = json.loads(response.text)   
 print(response.text)   
 image\_bytes = base64.b64decode(loads["image"]) # 转bytes   
 bytes\_io = io.BytesIO(image\_bytes) # 为了适应工具类 使用io进行封装   
 photo\_url = minioUtils.upload\_photo\_stream(bytes\_io, image\_name  
 return JsonResponse({'data': photo\_url, 'code': 200}, status=200)   
 return JsonResponse({'data': 'fail', 'code': 500}, status=500)   
@csrf\_exempt   
def pp\_photograph(request):   
 result, results, image\_name = TakePhotoForAIUtil.take\_photo\_pp(settings.CAP)   
 if result:   
 files = {'image': results}   
 # 请求模型   
 response = requests.post(settings.INTERFACE\_ADDRESS, files=files)   
 loads = json.loads(response.text)   
 print(loads["json\_data"])   
 image\_bytes = base64.b64decode(loads["image"]) # 转bytes   
 bytes\_io = io.BytesIO(image\_bytes) # 为了适应工具类 使用io进行封装   
 photo\_url = minioUtils.upload\_photo\_stream(bytes\_io, image\_nam  
 return JsonResponse({'data': photo\_url, 'code': 200}, status=200)   
 return JsonResponse({'data': 'fail', 'code': 500}, status=500)   
""""   
误差计算   
"""   
@csrf\_exempt   
def statistical\_errors(request):   
 # 读取json文件   
 json\_data = jsonUtil.read\_json\_file(settings.JSON\_FILE\_PATH)   
 error = linear\_straightness\_checker.calculate\_straightness\_error(   
 jsonUtil.query\_all\_issue\_data(json\_data, settings.DISPLACEMENTS), settings.LEVEL\_LENGTH)   
 # 清空 偏移数组   
 json\_data[settings.DISPLACEMENTS] = []   
 # 写入JSON文件   
 jsonUtil.write\_json\_file(settings.JSON\_FILE\_PATH, json\_data)   
 if error:   
 return JsonResponse({'data': error, 'code': 200}, status=200)   
 return JsonResponse({'data': 'fail', 'code': 500}, status=500)

# 4、 JSON database

{   
 "issueAtlas": [],   
 "correctAtlas": [],   
 "atlas": {   
 "issueAtlas": 4,   
 "correctAtlas": 13   
 },   
 "displacements": [   
 ]   
}

# 5、 HTML 模版

<!DOCTYPE html>   
<html lang="en">   
<head>   
 <meta charset="UTF-8">   
 <title>Title</title>   
</head>   
<body>   
<h1>上传图片</h1>   
<form id="uploadForm" enctype="multipart/form-data">   
 {% csrf\_token %} {# 如果您的后端使用了 CSRF 保护，请添加此行代码 #}   
 <input type="file" name="image" id="imageInput">   
 <button type="button" onclick="uploadImage()">上传</button>   
 <button type="button" onclick="openCamera()">开启摄像头</button>   
 <button type="button" onclick="photograph()">拍照</button>   
 <button type="button" onclick="pp\_photograph()">拍照 paddlepaddle离线模型</button>   
 <button type="button" onclick="closeCamera()">关闭摄像头</button>   
 <button type="button" onclick="statistical\_errors()">点击获取误差</button>   
</form>   
<h1>照片集合</h1>   
<div id="photoGallery">   
 <!-- 这里将用 JavaScript 动态添加照片 -->   
</div>   
<script>   
 function statistical\_errors() {   
 fetch('http://127.0.0.1:8000/statistical\_errors/', {   
 method: 'GET'   
 })   
 .then(response => response.json())   
 .then(data => {   
 console.log(data)   
 })   
 .catch(error => {   
 console.error('上传图片时出错：', error);   
 alert('上传图片时出错！');   
 });   
 }   
 function openCamera() {   
 fetch('http://127.0.0.1:8000/openCamera/', {   
 method: 'GET'   
 })   
 .then(response => response.json())   
 .then(data => {   
 console.log(data)   
 })   
 .catch(error => {   
 console.error('上传图片时出错：', error);   
 alert('上传图片时出错！');   
 });   
 }   
 function closeCamera() {   
 fetch('http://127.0.0.1:8000/closeCamera/', {   
 method: 'GET'   
 })   
 .then(response => response.json())   
 .then(data => {   
 console.log(data)   
 })   
 .catch(error => {   
 console.error('上传图片时出错：', error);   
 alert('上传图片时出错！');   
 });   
 }   
 function photograph() {   
   
 // const inputElement = document.getElementById('imageInput');   
 // if (inputElement.files.length === 0) {   
 // alert('请选择图片！');   
 // return;   
 // }  
 // const formData = new FormData();   
 // formData.append('image', inputElement.files[0]);   
 fetch('http://127.0.0.1:8000/photograph/', {   
 method: 'GET'   
 })   
 .then(response => response.json())   
 .then(data => {   
 console.log(data)   
 // 获取用于显示照片的容器元素   
 const galleryDiv = document.getElementById('photoGallery');   
   
 // 遍历照片 URL 数组并创建 img 元素来显示每个照片   
 data.data.forEach(data => {   
 const imgElement = document.createElement('img');   
 imgElement.src = data.photo\_url;   
 imgElement.alt = '照片';   
 const ScaleLineDistance\_spanElement = document.createElement('div');   
 ScaleLineDistance\_spanElement.innerHTML = '两根刻度线之间的距离为：' + data.ScaleLineDistance   
 const tickMarks\_center\_point\_spanElement = document.createElement('div');   
 tickMarks\_center\_point\_spanElement.innerHTML = '度线中心点为：' + data.tickMarks\_center\_point   
 const bubble\_placement\_spanElement = document.createElement('div');   
 bubble\_placement\_spanElement.innerHTML = '气泡在刻度线中心点的 ' + data.bubble\_placement + ' 第 ' + data.bubble\_left\_scale + ' 根线到第 ' + data.bubble\_right\_scale + ' 之间'   
 const bubbles\_distance\_half\_spanElement = document.createElement('div');   
 bubbles\_distance\_half\_spanElement.innerHTML = '气泡轮廓左侧到右侧的距离的二分之一为：' + data.bubbles\_distance\_half   
 const bubbles\_center\_point\_spanElement = document.createElement('div');   
 bubbles\_center\_point\_spanElement.innerHTML = '气泡中心点为：' + data.bubbles\_center\_point   
 const distance\_spanElement = document.createElement('div');   
 distance\_spanElement.innerHTML = '刻度线中心点和气泡中心点之间的距离为：' + data.distance   
 galleryDiv.appendChild(imgElement);   
 galleryDiv.appendChild(ScaleLineDistance\_spanElement);   
galleryDiv.appendChild(tickMarks\_center\_point\_spanElement);   
galleryDiv.appendChild(bubble\_placement\_spanElement);   
galleryDiv.appendChild(bubbles\_distance\_half\_spanElement);   
galleryDiv.appendChild(bubbles\_center\_point\_spanElement);   
 galleryDiv.appendChild(distance\_spanElement);   
 });   
 })   
 .catch(error => {   
 console.error('上传图片时出错：', error);   
 alert('上传图片时出错！');   
 });   
 }   
 function pp\_photograph() {   
 fetch('http://127.0.0.1:8000/pp\_photograph/', {   
 method: 'GET'   
 })   
 .then(response => response.json())   
 .then(data => {   
 console.log(data)   
 // 获取用于显示照片的容器元素   
 const galleryDiv = document.getElementById('photoGallery');   
 const imgElement = document.createElement('img');   
 imgElement.src = data.data;   
 imgElement.alt = '照片';   
 galleryDiv.appendChild(imgElement);   
 })   
 .catch(error => {   
 console.error('上传图片时出错：', error);   
 alert('上传图片时出错！');   
 });   
 }   
 function uploadImage() {   
 const inputElement = document.getElementById('imageInput');   
 if (inputElement.files.length === 0) {   
 alert('请选择图片！');   
 return;   
 }   
 const formData = new FormData();   
 formData.append('image', inputElement.files[0]);   
 fetch('http://127.0.0.1:8000/baidu/', {   
 method: 'POST',   
 body: formData   
 })   
 .then(response => response.json())   
 .then(data => {   
 console.log(data)   
 // 获取用于显示照片的容器元素   
 const galleryDiv = document.getElementById('photoGallery');   
 // 遍历照片 URL 数组并创建 img 元素来显示每个照片   
 data.data.forEach(data => {   
 const imgElement = document.createElement('img');   
 imgElement.src = data.photo\_url;   
 imgElement.alt = '照片';   
 const ScaleLineDistance\_spanElement = document.createElement('div');   
 ScaleLineDistance\_spanElement.innerHTML = '两根刻度线之间的距离为：' + data.ScaleLineDistance   
 const tickMarks\_center\_point\_spanElement = document.createElement('div');   
 tickMarks\_center\_point\_spanElement.innerHTML = '度线中心点为：' + data.tickMarks\_center\_point   
 const bubble\_placement\_spanElement = document.createElement('div');   
 bubble\_placement\_spanElement.innerHTML = '气泡在刻度线中心点的 ' + data.bubble\_placement + ' 第 ' + data.bubble\_left\_scale + ' 根线到第 ' + data.bubble\_right\_scale + ' 之间'   
 const bubbles\_distance\_half\_spanElement = document.createElement('div');   
 bubbles\_distance\_half\_spanElement.innerHTML = '气泡轮廓左侧到右侧的距离的二分之一为：' + data.bubbles\_distance\_half   
 const bubbles\_center\_point\_spanElement = document.createElement('div');   
 bubbles\_center\_point\_spanElement.innerHTML = '气泡中心点为：' + data.bubbles\_center\_point   
 const distance\_spanElement = document.createElement('div');   
 distance\_spanElement.innerHTML = '刻度线中心点和气泡中心点之间的距离为：' + data.distance   
 galleryDiv.appendChild(imgElement);   
galleryDiv.appendChild(ScaleLineDistance\_spanElement);   
galleryDiv.appendChild(tickMarks\_center\_point\_spanElement);   
galleryDiv.appendChild(bubble\_placement\_spanElement);   
galleryDiv.appendChild(bubbles\_distance\_half\_spanElement);   
galleryDiv.appendChild(bubbles\_center\_point\_spanElement);   
 galleryDiv.appendChild(distance\_spanElement);   
 });   
 })   
 .catch(error => {   
 console.error('上传图片时出错：', error);   
 alert('上传图片时出错！');   
 });   
 }   
</script>   
</body>   
</html>

# 6、前端

## 6.1 API

### 6.1.1 axios-instance.ts

import axios, {AxiosInstance} from 'axios';   
import {message} from "ant-design-vue";   
export const url = 'http://127.0.0.1:8000/'   
const axiosInstance: AxiosInstance = axios.create({   
 baseURL: url,   
 timeout: 60000,   
});   
// 添加请求拦截器   
axiosInstance.interceptors.request.use(   
 (config: any) => {   
 // 在发送请求之前做些什么   
 return config;   
 },   
 (error: any) => {   
 // 处理请求错误   
 return Promise.reject(error);   
 },   
);   
// 添加响应拦截器   
axiosInstance.interceptors.response.use(   
 (response: any) => {   
 const data = response.data   
 if (data.code != 200) {   
 message.error(data.msg)   
 // throw new Error(data.msg);   
 }   
 return data;   
 },   
 (error: any) => {   
 // 处理响应错误   
 return Promise.reject(error);   
 },   
);   
export default axiosInstance;

### 6.1.2 cameraApi.ts

import axiosInstance from "./axios-instance.ts";   
export default {   
 async openCamera() {   
 return axiosInstance.get('/openCamera')   
 },   
 async closeCamera() {   
 return axiosInstance.get('/closeCamera')   
 },   
 async takePhoto() {   
 return axiosInstance.get('/photograph')   
 }   
}

### 6.1.3 imgApi.ts

import axiosInstance from "./axios-instance.ts";   
type tableType = {   
 id ?: number,   
 img: string,   
 ScaleLineDistance: string,   
 tickMarksCenterPoint: string,   
 bubblePlacement: string,   
 bubblesDistanceHalf: string,   
 bubblesCenterPoint: string,   
 distance: string   
 createTime ?: string   
}   
export default {   
 async correctAtlas(img:tableType) {   
 return axiosInstance.post('/save/', {   
 atlas\_type: 'correctAtlas',   
 photo\_url: img.img,   
 ScaleLineDistance: img.ScaleLineDistance,   
 tickMarks\_center\_point: img.tickMarksCenterPoint,   
 bubble\_placement: img.bubblePlacement,   
 bubbles\_distance\_half: img.bubblesDistanceHalf,   
 bubbles\_center\_point: img.bubblesCenterPoint,   
 distance: img.distance,   
 createTime: img.createTime   
 })   
 },   
 async issueAtlas(img: any   
 ) {   
 return axiosInstance.post('/save/', {   
 atlas\_type: 'issueAtlas',   
 photo\_url: img.img,   
 ScaleLineDistance: img.ScaleLineDistance,   
 tickMarks\_center\_point: img.tickMarksCenterPoint,   
 bubble\_placement: img.bubblePlacement,   
 bubbles\_distance\_half: img.bubblesDistanceHalf,   
 bubbles\_center\_point: img.bubblesCenterPoint,   
 distance: img.distance,   
 createTime: img.createTime   
 })   
 },   
 async getImgs(type: string) {   
 return axiosInstance.get(`/get\_list/?atlas\_type=${type}`)   
 },   
 async deleteImg(id: number, atlasType: string) {   
 return axiosInstance.get(`/delete/?id=${id}&atlas\_type=${atlasType}`)   
 }   
}

## 6.2 store

import { defineStore } from 'pinia'   
import {ref} from "vue";   
// 你可以对 `defineStore()` 的返回值进行任意命名，但最好使用 store 的名字，同时以 `use` 开头且以 `Store` 结尾。(比如 `useUserStore`，`useCartStore`，`useProductStore`)   
// 第一个参数是你的应用中 Store 的唯一 ID。   
export const useResultStore = defineStore('result',() => {   
 const result = ref([])   
 const filterValue = ref('0')   
 const finishCount = ref(0)   
 return {result, filterValue, finishCount}   
})   
export const useIsUploadingStore = defineStore('isUpload', () => {   
 const isUploading = ref(false)   
 const progress = ref(0)   
 return {isUploading, progress}   
})

## 6.3 components

### 6.3.3 HelloWorld.vue

<template>   
 <a-layout class="layout" style="background-color: #f5f5f5;height: 100vh">   
 <a-layout-header style="background-color: #f5f5f5">   
 <div class="logo"/>   
 <a-row :gutter="16" justify="end">   
 <a-col>   
 <a-button @click="takePhoto" :loading="takePhotoLoading">   
 <template #icon>   
 <camera-outlined/>   
 </template>   
 拍张照片   
 </a-button>   
 </a-col>   
 <a-col style="display: flex; align-items: center">   
 <span>相机：</span>   
 <a-switch v-model:checked="checked" checked-children="开启" un-checked-children="关闭" :loading="switchLoading"   
 @change="cameraChange"/>   
 </a-col>   
 </a-row>   
 </a-layout-header>   
 <a-layout-content style="padding: 0 50px">   
 <div ref="content" :style="{ background: '#fff', padding: '24px', minHeight: '800px' }">   
 <a-menu   
 v-model:selectedKeys="selectedKeys"   
 mode="horizontal"   
 :style="{ lineHeight: '64px' }"   
 >   
 <a-menu-item key="1">首页</a-menu-item>   
 <a-menu-item key="2">审核</a-menu-item>   
 </a-menu>   
 <div v-if="selectedKeys[0] == 1" style="display: flex;justify-content: center;align-items: center;flex-direction: column;margin-top: 100px">   
 <uploader />   
 </div>   
 <resulter v-if="selectedKeys[0] == 2"/>   
 </div>   
 </a-layout-content>   
 <a-layout-footer style="text-align: center">   
 DS@2023   
 </a-layout-footer>   
 </a-layout>   
</template>   
<script lang="ts" setup>   
import {ref} from 'vue';   
import cameraApi from "../api/cameraApi.ts";   
import {message, UploadProps} from "ant-design-vue";   
import {CameraOutlined, InboxOutlined} from "@ant-design/icons-vue";   
import type {UploadChangeParam} from 'ant-design-vue';   
import axiosInstance from "../api/axios-instance.ts";   
import Uploader from "./uploader.vue";   
import Resulter from "./resulter.vue";   
import {useIsUploadingStore, useResultStore} from "../store/baiduResult.ts";   
import {storeToRefs} from "pinia";   
const takePhotoLoading = ref(false)   
const selectedKeys = ref<string[]>(['1']);   
const checked = ref(false)   
const switchLoading = ref(false)   
const baiduResult = useResultStore()   
const {result} = storeToRefs(baiduResult)   
const useIsUploading = useIsUploadingStore()   
const {isUploading} = storeToRefs(useIsUploading)   
const content = ref()   
const cameraChange = async (val) => {   
 if (val) {   
 switchLoading.value = true   
 let res   
 try {   
 res = await cameraApi.openCamera()   
 } catch (e) {   
 switchLoading.value = false   
 console.error(e)   
 message.error("未知错误")   
 return   
 }   
 if (res.code == 200)   
 message.success('打开成功！')   
 switchLoading.value = false   
 } else {   
 switchLoading.value = true   
 let res   
 try {   
 res = await cameraApi.closeCamera()   
 } catch (e) {   
 switchLoading.value = false   
 console.error(e)   
 message.error("未知错误")   
 return   
 }   
 if (res.code == 200)   
 message.success('已关闭！')   
 switchLoading.value = false   
 }   
}   
const takePhoto = async () => {   
 takePhotoLoading.value = true   
 message.loading("正在处理！")   
 let res;   
 try {   
 res = await cameraApi.takePhoto()   
 } catch (e) {   
 takePhotoLoading.value = false   
 message.error('未知错误')   
 return   
 }   
 if (res.code == 500) {   
 message.error('请检查相机是否打开!')   
 takePhotoLoading.value = false   
 return   
 }   
 const obj = res.data[0]   
 obj.createTime = new Date().toLocaleString()   
 result.value.push(obj)   
 takePhotoLoading.value = false   
 message.success('处理成功！')   
}   
</script>   
<style scoped>   
.logo {   
 float: left;   
 width: 120px;   
 height: 31px;   
 margin: 16px 24px 16px 0;   
 background: rgba(211, 211, 211, 0.71);   
}   
</style>

### 6.3.2 resulter.vue

<scrpt setup lang="ts">   
import {onMounted, ref, watch} from "vue";   
import {useResultStore} from "../store/baiduResult.ts";   
import {storeToRefs} from "pinia";   
import {message} from 'ant-design-vue';   
import imgApi from "../api/imgApi.ts";   
const columns = ref([   
 {   
 key: 'img',   
 title: '原图',   
 dataIndex: 'img',   
 resizable: true,   
 },   
 {   
 title: '两根刻度线之间的距离',   
 dataIndex: 'ScaleLineDistance',   
 resizable: true,   
 },   
 {   
 title: '刻度线中心点',   
 dataIndex: 'tickMarksCenterPoint',   
 resizable: true,   
 },   
 {   
 title: '气泡位置',   
 resizable: true,   
 dataIndex: 'bubblePlacement',   
 },   
 {   
 title: '气泡轮廓左侧到右侧的距离的二分之一',   
 dataIndex: 'bubblesDistanceHalf',   
 resizable: true,   
 },   
 {   
 title: '气泡中心点',   
 dataIndex: 'bubblesCenterPoint',   
 resizable: true,   
 },   
 {   
 title: '刻度线中心点和气泡中心点之间的距离',   
 resizable: true,   
 dataIndex: 'distance',   
 },   
 {   
 title: '上传时间',   
 resizable: true,   
 dataIndex: 'createTime',   
 },   
 {   
 title: '审核',   
 resizable: true,   
 key: 'edit',   
 dataIndex: 'edit',   
 }   
])   
   
function handleResizeColumn(w, col) {   
 col.width = w;   
}   
const tableData = ref([])   
const baiduResult = useResultStore()   
const {result, filterValue} = storeToRefs(baiduResult)   
watch(result.value, () => {   
 if (filterValue.value != '0') {   
 return   
 }   
 tableData.value = result.value.map(item => {   
 return {   
 key: Math.random(),   
 img: item.photo\_url,   
 ScaleLineDistance: Number(item.ScaleLineDistance).toFixed(3),   
 tickMarksCenterPoint: `${item.tickMarks\_center\_point}`,   
 bubblePlacement: `${item.bubble\_placement} 第 ${Number((item.bubble\_left\_scale)).toFixed(3)} 根线到第 ${Number((item.bubble\_right\_scale)).toFixed(3)} 之间`,   
 bubblesDistanceHalf: `${item.bubbles\_distance\_half}`,   
 bubblesCenterPoint: `${item.bubbles\_center\_point}`,   
 distance: Number(item.distance).toFixed(3),   
 createTime: item.createTime   
 }   
 })   
})   
onMounted(async () => {   
 await loadTableDate()   
})   
const confirm = async (img) => {   
 message.loading("正在提交，请稍后")   
 await imgApi.correctAtlas(img)   
 message.success('审核成功！')   
};   
   
const cancel = async (img) => {   
 message.loading("正在提交，请稍后")   
 await imgApi.issueAtlas(img)   
 message.success('审核成功！')   
};   
const loadTableDate = async () => {   
 if (filterValue.value == '0') {   
 tableData.value = result.value.map(item => {   
 return {   
 key: Math.random(),   
 img: item.photo\_url,   
 ScaleLineDistance: Number(item.ScaleLineDistance).toFixed(3),   
 tickMarksCenterPoint: `${item.tickMarks\_center\_point}`,   
 bubblePlacement: `${item.bubble\_placement} 第 ${Number((item.bubble\_left\_scale)).toFixed(3)} 根线到第 ${Number((item.bubble\_right\_scale)).toFixed(3)} 之间`,   
 bubblesDistanceHalf: `${item.bubbles\_distance\_half}`,   
 bubblesCenterPoint: `${item.bubbles\_center\_point}`,   
 distance: Number(item.distance).toFixed(3),   
 createTime: item.createTime   
 }   
 })   
 return   
 }   
 let resData = (await imgApi.getImgs(filterValue.value)).data   
 tableData.value = resData.map(item => ({   
 key: item.id,   
 id: item.id,   
 img: item.photo\_url,   
 ScaleLineDistance: Number(item.ScaleLineDistance).toFixed(3),   
 tickMarksCenterPoint: `${item.tickMarks\_center\_point}`,   
 bubblePlacement: item.bubble\_placement,   
 bubblesDistanceHalf: `${item.bubbles\_distance\_half}`,   
 bubblesCenterPoint: `${item.bubbles\_center\_point}`,   
 distance: Number(item.distance).toFixed(3),   
 createTime: item.createTime   
 }))   
}   
const termsChange =async () => {   
 await loadTableDate()   
}   
   
const deleted = async (img) => {   
 await imgApi.deleteImg(img.id, filterValue.value)   
 await loadTableDate()   
}   
const noDelete = () => {   
 return   
}   
</script>   
<template>   
 <div>   
 <a-radio-group v-model:value="filterValue" button-style="outline" @change="termsChange" style="margin: 10px 10px 10px 0">   
 <a-radio-button value="correctAtlas">合格</a-radio-button>   
 <a-radio-button value="issueAtlas">不合格</a-radio-button>   
 <a-radio-button value="0">全部</a-radio-button>   
 </a-radio-group>   
 </div>   
 <a-table :columns="columns" :data-source="tableData" @resizeColumn="handleResizeColumn" :scroll="{y: 600}" bordered   
 :pagination="false">   
 <template #bodyCell="{column, record}">   
 <template v-if="column.key === 'img'">   
 <a-image :src="record.img">   
 </a-image>   
 </template>   
 <template v-if="column.key === 'edit'">   
 <a-popconfirm   
 title="识别是否准确？"   
 ok-text="是"   
 cancel-text="否"   
 @confirm="confirm(record)"   
 @cancel="cancel(record)"   
 >   
 <a-button type="primary" ghost>审核</a-button>   
 </a-popconfirm>   
 <a-popconfirm   
 v-if="filterValue != 0"   
 title="是否删除？"   
 cancel-text="否"   
 ok-text="是"   
 @confirm="deleted(record)"   
 @cancel="noDelete()"   
 >   
 <a-button style="margin-left: 10px" danger>删除</a-button>   
 </a-popconfirm>   
 </template>   
 </template>   
 </a-table>   
</template>   
<style scoped>   
</style>

### 6.3.3 uploader.vue

<template>   
 <a-upload-dragger   
 style="width: 40vw"   
 v-model:fileList="fileList"   
 :before-upload="beforeUpload"   
 :multiple="true"   
 directory   
 @change="handleChange"   
 @drop="handleDrop"   
 :showUploadList="false"   
 >   
 <p class="ant-upload-drag-icon">   
 <inbox-outlined></inbox-outlined>   
 </p>   
 <p class="ant-upload-text">点击上传或将文件夹拖拽到此区域</p>   
 <p class="ant-upload-hint">   
 只支持上传图片   
 </p>   
 </a-upload-dragger>   
 <a-progress style="width: 50vw;margin-top: 20px" :percent="progress" v-show="isUploading"/>   
 <a-result   
 v-show="!isUploading && result.length > 0"   
 status="success"   
 title="任务已完成！"   
 >   
 </a-result>   
</template>   
<script lang="ts" setup>   
import {onMounted, ref, watch} from 'vue';   
import cameraApi from "../api/cameraApi.ts";   
import {message, UploadProps} from "ant-design-vue";   
import {CameraOutlined, InboxOutlined} from "@ant-design/icons-vue";   
import type {UploadChangeParam} from 'ant-design-vue';   
import axiosInstance from "../api/axios-instance.ts";   
import {storeToRefs} from "pinia";   
import {useIsUploadingStore, useResultStore} from "../store/baiduResult.ts";   
const baiduResult = useResultStore()   
const useIsUploading = useIsUploadingStore()   
const {isUploading, progress} = storeToRefs(useIsUploading)   
const fileList = ref([]);   
const {result, finishCount} = storeToRefs(baiduResult)   
watch(progress, () => {   
 if (progress.value >= 100) {   
 setTimeout(() => {   
 isUploading.value = false   
 finishCount.value = 0   
 progress.value = 0   
 fileList.value = []   
 }, 1000)   
 }   
})   
onMounted(() => {   
 if (progress.value >= 100) {   
 isUploading.value = false   
 finishCount.value = 0   
 progress.value = 0   
 fileList.value = []   
 }   
})   
const handleChange = (info: UploadChangeParam) => {   
 isUploading.value = true   
};   
function handleDrop(e: DragEvent) {   
 message.loading('开始上传')   
 isUploading.value = true   
}  
const beforeUpload: UploadProps['beforeUpload'] = file => {   
 const type = file.type   
 if (type == 'image/jpeg' || type == 'image/png' || type == 'image/jpg') {   
 fileList.value.push(file)   
 handleUpload(file)   
 }   
 return false;   
};   
const handleUpload = async (file) => {   
 const formData = new FormData();   
 formData.append('image', file as any);   
 let res = ''   
 try {   
 res = await axiosInstance.post('http://10.13.138.77:8000/baidu/',   
 formData   
 )   
 const obj = res.data[0]   
 obj.createTime = new Date().toLocaleString()   
 result.value.push(obj)   
 finishCount.value++   
 } catch (e) {   
 }   
 if (!res) {   
 result.value.push({   
 photo\_url: '123',   
 })   
 finishCount.value++   
 }   
 progress.value = Number(((finishCount.value/ fileList.value.length) \* 100).toFixed(2))   
};   
</script>   
<style scoped>   
</style>

## 6.4 APP

<script setup lang="ts">   
import HelloWorld from './components/HelloWorld.vue'   
</script>   
<template>   
 <HelloWorld />   
</template>   
<style scoped>   
</style>

## 6.5 MAIN

import { createApp } from 'vue'   
import './style.css'   
import App from './App.vue'   
import 'ant-design-vue/dist/reset.css';   
import Antd from 'ant-design-vue';   
import {createPinia} from "pinia";   
const app = createApp(App);   
const pinia = createPinia()   
app.use(pinia)   
app.use(Antd)   
app.mount('#app')

## 6.6 index

<!doctype html>   
<html lang="en">   
 <head>   
 <meta charset="UTF-8" />   
 <link rel="icon" type="image/svg+xml" href="/vite.svg" />   
 <meta name="viewport" content="width=device-width, initial-scale=1.0" />   
 <title>Vite + Vue + TS</title>   
 </head>   
 <body>   
 <div id="app"></div>   
 <script type="module" src="/src/main.ts"></script>   
 </body>   
</html>

## 6.7 package.json

{   
 "name": "frontend",   
 "private": true,   
 "version": "0.0.0",   
 "type": "module",   
 "scripts": {   
 "dev": "vite",   
 "build": "vue-tsc && vite build",   
 "preview": "vite preview"   
 },   
 "dependencies": {   
 "@ant-design/icons-vue": "^6.1.0",   
 "ant-design-vue": "4.x",   
 "axios": "^1.4.0",   
 "pinia": "^2.1.6",   
 "vue": "^3.3.4"   
 },   
 "devDependencies": {   
 "@vitejs/plugin-vue": "^4.2.3",   
 "typescript": "^5.0.2",   
 "vite": "^4.4.5",   
 "vue-tsc": "^1.8.5"   
 }   
}

## 6.8 package-lock.json

{   
 "name": "frontend",   
 "version": "0.0.0",   
 "lockfileVersion": 3,   
 "requires": true,   
 "packages": {   
 "": {   
 "name": "frontend",   
 "version": "0.0.0",   
 "dependencies": {   
 "@ant-design/icons-vue": "^6.1.0",   
 "ant-design-vue": "4.x",   
 "axios": "^1.4.0",   
 "pinia": "^2.1.6",   
 "vue": "^3.3.4"   
 },   
 "devDependencies": {   
 "@vitejs/plugin-vue": "^4.2.3",   
 "typescript": "^5.0.2",   
 "vite": "^4.4.5",   
 "vue-tsc": "^1.8.5"   
 }   
 },   
 "node\_modules/@ant-design/colors": {   
 "version": "6.0.0",   
 "resolved": "https://registry.npmmirror.com/@ant-design/colors/-/colors-6.0.0.tgz",   
 "integrity": "sha512-qAZRvPzfdWHtfameEGP2Qvuf838NhergR35o+EuVyB5XvSA98xod5r4utvi4TJ3ywmevm290g9nsCG5MryrdWQ==",   
 "license": "MIT",   
 "dependencies": {   
 "@ctrl/tinycolor": "^3.4.0"   
 }   
 },   
 "node\_modules/@ant-design/icons-svg": {   
 "version": "4.3.0",   
 "resolved": "https://registry.npmmirror.com/@ant-design/icons-svg/-/icons-svg-4.3.0.tgz",   
 "integrity": "sha512-WOgvdH/1Wl8Z7VXigRbCa5djO14zxrNTzvrAQzhWiBQtEKT0uTc8K1ltjKZ8U1gPn/wXhMA8/jE39SJl0WNxSg==",   
 "license": "MIT"   
 },   
 "node\_modules/@ant-design/icons-vue": {   
 "version": "6.1.0",   
 "resolved": "https://registry.npmmirror.com/@ant-design/icons-vue/-/icons-vue-6.1.0.tgz",   
 "integrity": "sha512-EX6bYm56V+ZrKN7+3MT/ubDkvJ5rK/O2t380WFRflDcVFgsvl3NLH7Wxeau6R8DbrO5jWR6DSTC3B6gYFp77AA==",   
 "dependencies": {   
 "@ant-design/colors": "^6.0.0",   
 "@ant-design/icons-svg": "^4.2.1"   
 },   
 "peerDependencies": {   
 "vue": ">=3.0.3"   
 }   
 },   
 "node\_modules/@babel/parser": {   
 "version": "7.22.7",   
 "resolved": "https://registry.npmmirror.com/@babel/parser/-/parser-7.22.7.tgz",   
 "integrity": "sha512-7NF8pOkHP5o2vpmGgNGcfAeCvOYhGLyA3Z4eBQkT1RJlWu47n63bCs93QfJ2hIAFCil7L5P2IWhs1oToVgrL0Q==",   
 "license": "MIT",   
 "bin": {   
 "parser": "bin/babel-parser.js"   
 },   
 "engines": {   
 "node": ">=6.0.0"   
 }   
 },   
 "node\_modules/@babel/runtime": {   
 "version": "7.22.6",   
 "resolved": "https://registry.npmmirror.com/@babel/runtime/-/runtime-7.22.6.tgz",   
 "integrity": "sha512-wDb5pWm4WDdF6LFUde3Jl8WzPA+3ZbxYqkC6xAXuD3irdEHN1k0NfTRrJD8ZD378SJ61miMLCqIOXYhd8x+AJQ==",   
 "license": "MIT",   
 "dependencies": {   
 "regenerator-runtime": "^0.13.11"   
 },   
 "engines": {   
 "node": ">=6.9.0"   
 }   
 },   
 "node\_modules/@ctrl/tinycolor": {   
 "version": "3.6.0",   
 "resolved": "https://registry.npmmirror.com/@ctrl/tinycolor/-/tinycolor-3.6.0.tgz",   
 "integrity": "sha512-/Z3l6pXthq0JvMYdUFyX9j0MaCltlIn6mfh9jLyQwg5aPKxkyNa0PTHtU1AlFXLNk55ZuAeJRcpvq+tmLfKmaQ==",   
 "license": "MIT",   
 "engines": {   
 "node": ">=10"   
 }   
 },   
 "node\_modules/@emotion/hash": {   
 "version": "0.9.1",   
 "resolved": "https://registry.npmmirror.com/@emotion/hash/-/hash-0.9.1.tgz",   
 "integrity": "sha512-gJB6HLm5rYwSLI6PQa+X1t5CFGrv1J1TWG+sOyMCeKz2ojaj6Fnl/rZEspogG+cvqbt4AE/2eIyD2QfLKTBNlQ==",   
 "license": "MIT"   
 },   
 "node\_modules/@emotion/unitless": {   
 "version": "0.8.1",   
 "resolved": "https://registry.npmmirror.com/@emotion/unitless/-/unitless-0.8.1.tgz",   
 "integrity": "sha512-KOEGMu6dmJZtpadb476IsZBclKvILjopjUii3V+7MnXIQCYh8W3NgNcgwo21n9LXZX6EDIKvqfjYxXebDwxKmQ==",   
 "license": "MIT"   
 },   
 "node\_modules/@esbuild/win32-x64": {   
 "version": "0.18.17",   
 "resolved": "https://registry.npmmirror.com/@esbuild/win32-x64/-/win32-x64-0.18.17.tgz",   
 "integrity": "sha512-fGEb8f2BSA3CW7riJVurug65ACLuQAzKq0SSqkY2b2yHHH0MzDfbLyKIGzHwOI/gkHcxM/leuSW6D5w/LMNitA==",   
 "cpu": [   
 "x64"   
 ],   
 "dev": true,   
 "license": "MIT",   
 "optional": true,   
 "os": [   
 "win32"   
 ],   
 "engines": {   
 "node": ">=12"   
 }   
 },   
 "node\_modules/@jridgewell/sourcemap-codec": {   
 "version": "1.4.15",   
 "resolved": "https://registry.npmmirror.com/@jridgewell/sourcemap-codec/-/sourcemap-codec-1.4.15.tgz",   
 "integrity": "sha512-eF2rxCRulEKXHTRiDrDy6erMYWqNw4LPdQ8UQA4huuxaQsVeRPFl2oM8oDGxMFhJUWZf9McpLtJasDDZb/Bpeg==",   
 "license": "MIT"   
 },   
 "node\_modules/@simonwep/pickr": {   
 "version": "1.8.2",   
 "resolved": "https://registry.npmmirror.com/@simonwep/pickr/-/pickr-1.8.2.tgz",   
 "integrity": "sha512-/l5w8BIkrpP6n1xsetx9MWPWlU6OblN5YgZZphxan0Tq4BByTCETL6lyIeY8lagalS2Nbt4F2W034KHLIiunKA==",   
 "license": "MIT",   
 "dependencies": {   
 "core-js": "^3.15.1",   
 "nanopop": "^2.1.0"   
 }   
 },   
 "node\_modules/@vitejs/plugin-vue": {   
 "version": "4.2.3",   
 "resolved": "https://registry.npmmirror.com/@vitejs/plugin-vue/-/plugin-vue-4.2.3.tgz",   
 "integrity": "sha512-R6JDUfiZbJA9cMiguQ7jxALsgiprjBeHL5ikpXfJCH62pPHtI+JdJ5xWj6Ev73yXSlYl86+blXn1kZHQ7uElxw==",   
 "dev": true,   
 "license": "MIT",   
 "engines": {   
 "node": "^14.18.0 || >=16.0.0"   
 },   
 "peerDependencies": {   
 "vite": "^4.0.0",   
 "vue": "^3.2.25"   
 }   
 },   
 "node\_modules/@volar/language-core": {   
 "version": "1.10.0",   
 "resolved": "https://registry.npmmirror.com/@volar/language-core/-/language-core-1.10.0.tgz",   
 "integrity": "sha512-ddyWwSYqcbEZNFHm+Z3NZd6M7Ihjcwl/9B5cZd8kECdimVXUFdFi60XHWD27nrWtUQIsUYIG7Ca1WBwV2u2LSQ==",   
 "dev": true,   
 "license": "MIT",   
 "dependencies": {   
 "@volar/source-map": "1.10.0"   
 }   
 },   
 "node\_modules/@volar/source-map": {   
 "version": "1.10.0",   
 "resolved": "https://registry.npmmirror.com/@volar/source-map/-/source-map-1.10.0.tgz",   
 "integrity": "sha512-/ibWdcOzDGiq/GM1JU2eX8fH1bvAhl66hfe8yEgLEzg9txgr6qb5sQ/DEz5PcDL75tF5H5sCRRwn8Eu8ezi9mw==",   
 "dev": true,   
 "license": "MIT",   
 "dependencies": {   
 "muggle-string": "^0.3.1"   
 }   
 },   
 "node\_modules/@volar/typescript": {   
 "version": "1.10.0",   
 "resolved": "https://registry.npmmirror.com/@volar/typescript/-/typescript-1.10.0.tgz",   
 "integrity": "sha512-OtqGtFbUKYC0pLNIk3mHQp5xWnvL1CJIUc9VE39VdZ/oqpoBh5jKfb9uJ45Y4/oP/WYTrif/Uxl1k8VTPz66Gg==",   
 "dev": true,   
 "license": "MIT",   
 "dependencies": {   
 "@volar/language-core": "1.10.0"   
 }   
 },   
 "node\_modules/@vue/compiler-core": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/compiler-core/-/compiler-core-3.3.4.tgz",   
 "integrity": "sha512-cquyDNvZ6jTbf/+x+AgM2Arrp6G4Dzbb0R64jiG804HRMfRiFXWI6kqUVqZ6ZR0bQhIoQjB4+2bhNtVwndW15g==",   
 "license": "MIT",   
 "dependencies": {   
 "@babel/parser": "^7.21.3",   
 "@vue/shared": "3.3.4",   
 "estree-walker": "^2.0.2",   
 "source-map-js": "^1.0.2"   
 }   
 },   
 "node\_modules/@vue/compiler-dom": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/compiler-dom/-/compiler-dom-3.3.4.tgz",   
 "integrity": "sha512-wyM+OjOVpuUukIq6p5+nwHYtj9cFroz9cwkfmP9O1nzH68BenTTv0u7/ndggT8cIQlnBeOo6sUT/gvHcIkLA5w==",   
 "license": "MIT",   
 "dependencies": {   
 "@vue/compiler-core": "3.3.4",   
 "@vue/shared": "3.3.4"   
 }   
 },   
 "node\_modules/@vue/compiler-sfc": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/compiler-sfc/-/compiler-sfc-3.3.4.tgz",   
 "integrity": "sha512-6y/d8uw+5TkCuzBkgLS0v3lSM3hJDntFEiUORM11pQ/hKvkhSKZrXW6i69UyXlJQisJxuUEJKAWEqWbWsLeNKQ==",   
 "license": "MIT",   
 "dependencies": {   
 "@babel/parser": "^7.20.15",   
 "@vue/compiler-core": "3.3.4",   
 "@vue/compiler-dom": "3.3.4",   
 "@vue/compiler-ssr": "3.3.4",   
 "@vue/reactivity-transform": "3.3.4",   
 "@vue/shared": "3.3.4",   
 "estree-walker": "^2.0.2",   
 "magic-string": "^0.30.0",   
 "postcss": "^8.1.10",   
 "source-map-js": "^1.0.2"   
 }   
 },   
 "node\_modules/@vue/compiler-ssr": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/compiler-ssr/-/compiler-ssr-3.3.4.tgz",   
 "integrity": "sha512-m0v6oKpup2nMSehwA6Uuu+j+wEwcy7QmwMkVNVfrV9P2qE5KshC6RwOCq8fjGS/Eak/uNb8AaWekfiXxbBB6gQ==",   
 "license": "MIT",   
 "dependencies": {   
 "@vue/compiler-dom": "3.3.4",   
 "@vue/shared": "3.3.4"   
 }   
 },   
 "node\_modules/@vue/devtools-api": {   
 "version": "6.5.0",   
 "resolved": "https://registry.npmmirror.com/@vue/devtools-api/-/devtools-api-6.5.0.tgz",   
 "integrity": "sha512-o9KfBeaBmCKl10usN4crU53fYtC1r7jJwdGKjPT24t348rHxgfpZ0xL3Xm/gLUYnc0oTp8LAmrxOeLyu6tbk2Q=="   
 },   
 "node\_modules/@vue/language-core": {   
 "version": "1.8.8",   
 "resolved": "https://registry.npmmirror.com/@vue/language-core/-/language-core-1.8.8.tgz",   
 "integrity": "sha512-i4KMTuPazf48yMdYoebTkgSOJdFraE4pQf0B+FTOFkbB+6hAfjrSou/UmYWRsWyZV6r4Rc6DDZdI39CJwL0rWw==",   
 "dev": true,   
 "license": "MIT",   
 "dependencies": {   
 "@volar/language-core": "~1.10.0",   
 "@volar/source-map": "~1.10.0",   
 "@vue/compiler-dom": "^3.3.0",   
 "@vue/reactivity": "^3.3.0",   
 "@vue/shared": "^3.3.0",   
 "minimatch": "^9.0.0",   
 "muggle-string": "^0.3.1",   
 "vue-template-compiler": "^2.7.14"   
 },   
 "peerDependencies": {   
 "typescript": "\*"   
 },   
 "peerDependenciesMeta": {   
 "typescript": {   
 "optional": true   
 }   
 }   
 },   
 "node\_modules/@vue/reactivity": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/reactivity/-/reactivity-3.3.4.tgz",   
 "integrity": "sha512-kLTDLwd0B1jG08NBF3R5rqULtv/f8x3rOFByTDz4J53ttIQEDmALqKqXY0J+XQeN0aV2FBxY8nJDf88yvOPAqQ==",   
 "license": "MIT",   
 "dependencies": {   
 "@vue/shared": "3.3.4"   
 }   
 },   
 "node\_modules/@vue/reactivity-transform": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/reactivity-transform/-/reactivity-transform-3.3.4.tgz",   
 "integrity": "sha512-MXgwjako4nu5WFLAjpBnCj/ieqcjE2aJBINUNQzkZQfzIZA4xn+0fV1tIYBJvvva3N3OvKGofRLvQIwEQPpaXw==",   
 "license": "MIT",   
 "dependencies": {   
 "@babel/parser": "^7.20.15",   
 "@vue/compiler-core": "3.3.4",   
 "@vue/shared": "3.3.4",   
 "estree-walker": "^2.0.2",   
 "magic-string": "^0.30.0"   
 }   
 },   
 "node\_modules/@vue/runtime-core": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/runtime-core/-/runtime-core-3.3.4.tgz",   
 "integrity": "sha512-R+bqxMN6pWO7zGI4OMlmvePOdP2c93GsHFM/siJI7O2nxFRzj55pLwkpCedEY+bTMgp5miZ8CxfIZo3S+gFqvA==",   
 "license": "MIT",   
 "dependencies": {   
 "@vue/reactivity": "3.3.4",   
 "@vue/shared": "3.3.4"   
 }   
 },   
 "node\_modules/@vue/runtime-dom": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/runtime-dom/-/runtime-dom-3.3.4.tgz",   
 "integrity": "sha512-Aj5bTJ3u5sFsUckRghsNjVTtxZQ1OyMWCr5dZRAPijF/0Vy4xEoRCwLyHXcj4D0UFbJ4lbx3gPTgg06K/GnPnQ==",   
 "license": "MIT",   
 "dependencies": {   
 "@vue/runtime-core": "3.3.4",   
 "@vue/shared": "3.3.4",   
 "csstype": "^3.1.1"   
 }   
 },   
 "node\_modules/@vue/server-renderer": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/server-renderer/-/server-renderer-3.3.4.tgz",   
 "integrity": "sha512-Q6jDDzR23ViIb67v+vM1Dqntu+HUexQcsWKhhQa4ARVzxOY2HbC7QRW/ggkDBd5BU+uM1sV6XOAP0b216o34JQ==",   
 "license": "MIT",   
 "dependencies": {   
 "@vue/compiler-ssr": "3.3.4",   
 "@vue/shared": "3.3.4"   
 },   
 "peerDependencies": {   
 "vue": "3.3.4"   
 }   
 },   
 "node\_modules/@vue/shared": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/shared/-/shared-3.3.4.tgz",   
 "integrity": "sha512-7OjdcV8vQ74eiz1TZLzZP4JwqM5fA94K6yntPS5Z25r9HDuGNzaGdgvwKYq6S+MxwF0TFRwe50fIR/MYnakdkQ==",   
 "license": "MIT"   
 },   
 "node\_modules/@vue/typescript": {   
 "version": "1.8.8",   
 "resolved": "https://registry.npmmirror.com/@vue/typescript/-/typescript-1.8.8.tgz",   
 "integrity": "sha512-jUnmMB6egu5wl342eaUH236v8tdcEPXXkPgj+eI/F6JwW/lb+yAU6U07ZbQ3MVabZRlupIlPESB7ajgAGixhow==",   
 "dev": true,   
 "license": "MIT",   
 "dependencies": {   
 "@volar/typescript": "~1.10.0",   
 "@vue/language-core": "1.8.8"   
 }   
 },   
 "node\_modules/ant-design-vue": {   
 "version": "4.0.0",   
 "resolved": "https://registry.npmmirror.com/ant-design-vue/-/ant-design-vue-4.0.0.tgz",   
 "integrity": "sha512-6V8DxHPv3HJYDdv48W/KFOUjeZDsYJhFlk13HFSKCJSR2MQs4YnPIAC+rWie9vYcVEKbO+cQDqiOM3cBoHMNUQ==",   
 "license": "MIT",   
 "dependencies": {   
 "@ant-design/colors": "^6.0.0",   
 "@ant-design/icons-vue": "^6.1.0",   
 "@babel/runtime": "^7.10.5",   
 "@ctrl/tinycolor": "^3.5.0",   
 "@emotion/hash": "^0.9.0",   
 "@emotion/unitless": "^0.8.0",   
 "@simonwep/pickr": "~1.8.0",   
 "array-tree-filter": "^2.1.0",   
 "async-validator": "^4.0.0",   
 "csstype": "^3.1.1",   
 "dayjs": "^1.10.5",   
 "dom-align": "^1.12.1",   
 "dom-scroll-into-view": "^2.0.0",   
 "lodash": "^4.17.21",   
 "lodash-es": "^4.17.15",   
 "resize-observer-polyfill": "^1.5.1",   
 "scroll-into-view-if-needed": "^2.2.25",   
 "shallow-equal": "^1.0.0",   
 "stylis": "^4.1.3",   
 "throttle-debounce": "^5.0.0",   
 "vue-types": "^3.0.0",   
 "warning": "^4.0.0"   
 },   
 "engines": {   
 "node": ">=12.22.0"   
 },   
 "funding": {   
 "type": "opencollective",   
 "url": "https://opencollective.com/ant-design-vue"   
 },   
 "peerDependencies": {   
 "vue": ">=3.2.0"   
 }   
 },   
 "node\_modules/array-tree-filter": {   
 "version": "2.1.0",   
 "resolved": "https://registry.npmmirror.com/array-tree-filter/-/array-tree-filter-2.1.0.tgz",   
 "integrity": "sha512-4ROwICNlNw/Hqa9v+rk5h22KjmzB1JGTMVKP2AKJBOCgb0yL0ASf0+YvCcLNNwquOHNX48jkeZIJ3a+oOQqKcw==",   
 "license": "MIT"   
 },   
 "node\_modules/async-validator": {   
 "version": "4.2.5",   
 "resolved": "https://registry.npmmirror.com/async-validator/-/async-validator-4.2.5.tgz",   
 "integrity": "sha512-7HhHjtERjqlNbZtqNqy2rckN/SpOOlmDliet+lP7k+eKZEjPk3DgyeU9lIXLdeLz0uBbbVp+9Qdow9wJWgwwfg==",   
 "license": "MIT"   
 },   
 "node\_modules/asynckit": {   
 "version": "0.4.0",   
 "resolved": "https://registry.npmmirror.com/asynckit/-/asynckit-0.4.0.tgz",   
 "integrity": "sha512-Oei9OH4tRh0YqU3GxhX79dM/mwVgvbZJaSNaRk+bshkj0S5cfHcgYakreBjrHwatXKbz+IoIdYLxrKim2MjW0Q==",   
 "license": "MIT"   
 },   
 "node\_modules/axios": {   
 "version": "1.4.0",   
 "resolved": "https://registry.npmmirror.com/axios/-/axios-1.4.0.tgz",   
 "integrity": "sha512-S4XCWMEmzvo64T9GfvQDOXgYRDJ/wsSZc7Jvdgx5u1sd0JwsuPLqb3SYmusag+edF6ziyMensPVqLTSc1PiSEA==",   
 "license": "MIT",   
 "dependencies": {   
 "follow-redirects": "^1.15.0",   
 "form-data": "^4.0.0",   
 "proxy-from-env": "^1.1.0"   
 }   
 },   
 "node\_modules/balanced-match": {   
 "version": "1.0.2",   
 "resolved": "https://registry.npmmirror.com/balanced-match/-/balanced-match-1.0.2.tgz",   
 "integrity": "sha512-3oSeUO0TMV67hN1AmbXsK4yaqU7tjiHlbxRDZOpH0KW9+CeX4bRAaX0Anxt0tx2MrpRpWwQaPwIlISEJhYU5Pw==",   
 "dev": true,   
 "license": "MIT"   
 },   
 "node\_modules/brace-expansion": {   
 "version": "2.0.1",   
 "resolved": "https://registry.npmmirror.com/brace-expansion/-/brace-expansion-2.0.1.tgz",   
 "integrity": "sha512-XnAIvQ8eM+kC6aULx6wuQiwVsnzsi9d3WxzV3FpWTGA19F621kwdbsAcFKXgKUHZWsy+mY6iL1sHTxWEFCytDA==",   
 "dev": true,   
 "license": "MIT",   
 "dependencies": {   
 "balanced-match": "^1.0.0"   
 }   
 },   
 "node\_modules/combined-stream": {   
 "version": "1.0.8",

}   
 },   
 "node\_modules/@vue/compiler-ssr": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/compiler-ssr/-/compiler-ssr-3.3.4.tgz",   
 "integrity": "sha512-m0v6oKpup2nMSehwA6Uuu+j+wEwcy7QmwMkVNVfrV9P2qE5KshC6RwOCq8fjGS/Eak/uNb8AaWekfiXxbBB6gQ==",   
 "license": "MIT",   
 "dependencies": {   
 "@vue/compiler-dom": "3.3.4",   
 "@vue/shared": "3.3.4"   
 }   
 },   
 "node\_modules/@vue/devtools-api": {   
 "version": "6.5.0",   
 "resolved": "https://registry.npmmirror.com/@vue/devtools-api/-/devtools-api-6.5.0.tgz",   
 "integrity": "sha512-o9KfBeaBmCKl10usN4crU53fYtC1r7jJwdGKjPT24t348rHxgfpZ0xL3Xm/gLUYnc0oTp8LAmrxOeLyu6tbk2Q=="   
 },   
 "node\_modules/@vue/language-core": {   
 "version": "1.8.8",   
 "resolved": "https://registry.npmmirror.com/@vue/language-core/-/language-core-1.8.8.tgz",   
 "integrity": "sha512-i4KMTuPazf48yMdYoebTkgSOJdFraE4pQf0B+FTOFkbB+6hAfjrSou/UmYWRsWyZV6r4Rc6DDZdI39CJwL0rWw==",   
 "dev": true,   
 "license": "MIT",   
 "dependencies": {   
 "@volar/language-core": "~1.10.0",   
 "@volar/source-map": "~1.10.0",   
 "@vue/compiler-dom": "^3.3.0",   
 "@vue/reactivity": "^3.3.0",   
 "@vue/shared": "^3.3.0",   
 "minimatch": "^9.0.0",   
 "muggle-string": "^0.3.1",   
 "vue-template-compiler": "^2.7.14"   
 },   
 "peerDependencies": {   
 "typescript": "\*"   
 },   
 "peerDependenciesMeta": {   
 "typescript": {   
 "optional": true   
 }   
 }   
 },   
 "node\_modules/@vue/reactivity": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/reactivity/-/reactivity-3.3.4.tgz",   
 "integrity": "sha512-kLTDLwd0B1jG08NBF3R5rqULtv/f8x3rOFByTDz4J53ttIQEDmALqKqXY0J+XQeN0aV2FBxY8nJDf88yvOPAqQ==",   
 "license": "MIT",   
 "dependencies": {   
 "@vue/shared": "3.3.4"   
 }   
 },   
 "node\_modules/@vue/reactivity-transform": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/reactivity-transform/-/reactivity-transform-3.3.4.tgz",   
 "integrity": "sha512-MXgwjako4nu5WFLAjpBnCj/ieqcjE2aJBINUNQzkZQfzIZA4xn+0fV1tIYBJvvva3N3OvKGofRLvQIwEQPpaXw==",   
 "license": "MIT",   
 "dependencies": {   
 "@babel/parser": "^7.20.15",   
 "@vue/compiler-core": "3.3.4",   
 "@vue/shared": "3.3.4",   
 "estree-walker": "^2.0.2",   
 "magic-string": "^0.30.0"   
 }   
 },   
 "node\_modules/@vue/runtime-core": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/runtime-core/-/runtime-core-3.3.4.tgz",   
 "integrity": "sha512-R+bqxMN6pWO7zGI4OMlmvePOdP2c93GsHFM/siJI7O2nxFRzj55pLwkpCedEY+bTMgp5miZ8CxfIZo3S+gFqvA==",   
 "license": "MIT",   
 "dependencies": {   
 "@vue/reactivity": "3.3.4",   
 "@vue/shared": "3.3.4"   
 }   
 },   
 "node\_modules/@vue/runtime-dom": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/runtime-dom/-/runtime-dom-3.3.4.tgz",   
 "integrity": "sha512-Aj5bTJ3u5sFsUckRghsNjVTtxZQ1OyMWCr5dZRAPijF/0Vy4xEoRCwLyHXcj4D0UFbJ4lbx3gPTgg06K/GnPnQ==",   
 "license": "MIT",   
 "dependencies": {   
 "@vue/runtime-core": "3.3.4",   
 "@vue/shared": "3.3.4",   
 "csstype": "^3.1.1"   
 }   
 },   
 "node\_modules/@vue/server-renderer": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/server-renderer/-/server-renderer-3.3.4.tgz",   
 "integrity": "sha512-Q6jDDzR23ViIb67v+vM1Dqntu+HUexQcsWKhhQa4ARVzxOY2HbC7QRW/ggkDBd5BU+uM1sV6XOAP0b216o34JQ==",   
 "license": "MIT",   
 "dependencies": {   
 "@vue/compiler-ssr": "3.3.4",   
 "@vue/shared": "3.3.4"   
 },   
 "peerDependencies": {   
 "vue": "3.3.4"   
 }   
 },   
 "node\_modules/@vue/shared": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/shared/-/shared-3.3.4.tgz",   
 "integrity": "sha512-7OjdcV8vQ74eiz1TZLzZP4JwqM5fA94K6yntPS5Z25r9HDuGNzaGdgvwKYq6S+MxwF0TFRwe50fIR/MYnakdkQ==",   
 "license": "MIT"   
 },   
 "node\_modules/@vue/typescript": {   
 "version": "1.8.8",   
 "resolved": "https://registry.npmmirror.com/@vue/typescript/-/typescript-1.8.8.tgz",   
 "integrity": "sha512-jUnmMB6egu5wl342eaUH236v8tdcEPXXkPgj+eI/F6JwW/lb+yAU6U07ZbQ3MVabZRlupIlPESB7ajgAGixhow==",   
 "dev": true,   
 "license": "MIT",   
 "dependencies": {   
 "@volar/typescript": "~1.10.0",   
 "@vue/language-core": "1.8.8"   
 }   
 },   
 "node\_modules/ant-design-vue": {   
 "version": "4.0.0",   
 "resolved": "https://registry.npmmirror.com/ant-design-vue/-/ant-design-vue-4.0.0.tgz",   
 "integrity": "sha512-6V8DxHPv3HJYDdv48W/KFOUjeZDsYJhFlk13HFSKCJSR2MQs4YnPIAC+rWie9vYcVEKbO+cQDqiOM3cBoHMNUQ==",   
 "license": "MIT",   
 "dependencies": {   
 "@ant-design/colors": "^6.0.0",   
 "@ant-design/icons-vue": "^6.1.0",   
 "@babel/runtime": "^7.10.5",   
 "@ctrl/tinycolor": "^3.5.0",   
 "@emotion/hash": "^0.9.0",   
 "@emotion/unitless": "^0.8.0",   
 "@simonwep/pickr": "~1.8.0",   
 "array-tree-filter": "^2.1.0",   
 "async-validator": "^4.0.0",   
 "csstype": "^3.1.1",   
 "dayjs": "^1.10.5",   
 "dom-align": "^1.12.1",   
 "dom-scroll-into-view": "^2.0.0",   
 "lodash": "^4.17.21",   
 "lodash-es": "^4.17.15",   
 "resize-observer-polyfill": "^1.5.1",   
 "scroll-into-view-if-needed": "^2.2.25",   
 "shallow-equal": "^1.0.0",   
 "stylis": "^4.1.3",   
 "throttle-debounce": "^5.0.0",   
 "vue-types": "^3.0.0",   
 "warning": "^4.0.0"   
 },   
 "engines": {   
 "node": ">=12.22.0"   
 },   
 "funding": {   
 "type": "opencollective",   
 "url": "https://opencollective.com/ant-design-vue"   
 },   
 "peerDependencies": {   
 "vue": ">=3.2.0"   
 }   
 },   
 "node\_modules/array-tree-filter": {   
 "version": "2.1.0",   
 "resolved": "https://registry.npmmirror.com/array-tree-filter/-/array-tree-filter-2.1.0.tgz",   
 "integrity": "sha512-4ROwICNlNw/Hqa9v+rk5h22KjmzB1JGTMVKP2AKJBOCgb0yL0ASf0+YvCcLNNwquOHNX48jkeZIJ3a+oOQqKcw==",   
 "license": "MIT"   
 },   
 "node\_modules/async-validator": {   
 "version": "4.2.5",   
 "resolved": "https://registry.npmmirror.com/async-validator/-/async-validator-4.2.5.tgz",   
 "integrity": "sha512-7HhHjtERjqlNbZtqNqy2rckN/SpOOlmDliet+lP7k+eKZEjPk3DgyeU9lIXLdeLz0uBbbVp+9Qdow9wJWgwwfg==",   
 "license": "MIT"   
 },   
 "node\_modules/asynckit": {   
 "version": "0.4.0",   
 "resolved": "https://registry.npmmirror.com/asynckit/-/asynckit-0.4.0.tgz",   
 "integrity": "sha512-Oei9OH4tRh0YqU3GxhX79dM/mwVgvbZJaSNaRk+bshkj0S5cfHcgYakreBjrHwatXKbz+IoIdYLxrKim2MjW0Q==",   
 "license": "MIT"   
 },   
 "node\_modules/axios": {   
 "version": "1.4.0",   
 "resolved": "https://registry.npmmirror.com/axios/-/axios-1.4.0.tgz",   
 "integrity": "sha512-S4XCWMEmzvo64T9GfvQDOXgYRDJ/wsSZc7Jvdgx5u1sd0JwsuPLqb3SYmusag+edF6ziyMensPVqLTSc1PiSEA==",   
 "license": "MIT",   
 "dependencies": {   
 "follow-redirects": "^1.15.0",   
 "form-data": "^4.0.0",   
 "proxy-from-env": "^1.1.0"   
 }   
 },   
 "node\_modules/balanced-match": {   
 "version": "1.0.2",   
 "resolved": "https://registry.npmmirror.com/balanced-match/-/balanced-match-1.0.2.tgz",   
 "integrity": "sha512-3oSeUO0TMV67hN1AmbXsK4yaqU7tjiHlbxRDZOpH0KW9+CeX4bRAaX0Anxt0tx2MrpRpWwQaPwIlISEJhYU5Pw==",   
 "dev": true,   
 "license": "MIT"   
 },   
 "node\_modules/brace-expansion": {   
 "version": "2.0.1",   
 "resolved": "https://registry.npmmirror.com/brace-expansion/-/brace-expansion-2.0.1.tgz",   
 "integrity": "sha512-XnAIvQ8eM+kC6aULx6wuQiwVsnzsi9d3WxzV3FpWTGA19F621kwdbsAcFKXgKUHZWsy+mY6iL1sHTxWEFCytDA==",   
 "dev": true,   
 "license": "MIT",   
 "dependencies": {   
 "balanced-match": "^1.0.0"   
 }   
 },   
 "node\_modules/combined-stream": {   
 "version": "1.0.8", }   
 },   
 "node\_modules/@vue/compiler-ssr": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/compiler-ssr/-/compiler-ssr-3.3.4.tgz",   
 "integrity": "sha512-m0v6oKpup2nMSehwA6Uuu+j+wEwcy7QmwMkVNVfrV9P2qE5KshC6RwOCq8fjGS/Eak/uNb8AaWekfiXxbBB6gQ==",   
 "license": "MIT",   
 "dependencies": {   
 "@vue/compiler-dom": "3.3.4",   
 "@vue/shared": "3.3.4"   
 }   
 },   
 "node\_modules/@vue/devtools-api": {   
 "version": "6.5.0",   
 "resolved": "https://registry.npmmirror.com/@vue/devtools-api/-/devtools-api-6.5.0.tgz",   
 "integrity": "sha512-o9KfBeaBmCKl10usN4crU53fYtC1r7jJwdGKjPT24t348rHxgfpZ0xL3Xm/gLUYnc0oTp8LAmrxOeLyu6tbk2Q=="   
 },   
 "node\_modules/@vue/language-core": {   
 "version": "1.8.8",   
 "resolved": "https://registry.npmmirror.com/@vue/language-core/-/language-core-1.8.8.tgz",   
 "integrity": "sha512-i4KMTuPazf48yMdYoebTkgSOJdFraE4pQf0B+FTOFkbB+6hAfjrSou/UmYWRsWyZV6r4Rc6DDZdI39CJwL0rWw==",   
 "dev": true,   
 "license": "MIT",   
 "dependencies": {   
 "@volar/language-core": "~1.10.0",   
 "@volar/source-map": "~1.10.0",   
 "@vue/compiler-dom": "^3.3.0",   
 "@vue/reactivity": "^3.3.0",   
 "@vue/shared": "^3.3.0",   
 "minimatch": "^9.0.0",   
 "muggle-string": "^0.3.1",   
 "vue-template-compiler": "^2.7.14"   
 },   
 "peerDependencies": {   
 "typescript": "\*"   
 },   
 "peerDependenciesMeta": {   
 "typescript": {   
 "optional": true   
 }   
 }   
 },   
 "node\_modules/@vue/reactivity": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/reactivity/-/reactivity-3.3.4.tgz",   
 "integrity": "sha512-kLTDLwd0B1jG08NBF3R5rqULtv/f8x3rOFByTDz4J53ttIQEDmALqKqXY0J+XQeN0aV2FBxY8nJDf88yvOPAqQ==",   
 "license": "MIT",   
 "dependencies": {   
 "@vue/shared": "3.3.4"   
 }   
 },   
 "node\_modules/@vue/reactivity-transform": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/reactivity-transform/-/reactivity-transform-3.3.4.tgz",   
 "integrity": "sha512-MXgwjako4nu5WFLAjpBnCj/ieqcjE2aJBINUNQzkZQfzIZA4xn+0fV1tIYBJvvva3N3OvKGofRLvQIwEQPpaXw==",   
 "license": "MIT",   
 "dependencies": {   
 "@babel/parser": "^7.20.15",   
 "@vue/compiler-core": "3.3.4",   
 "@vue/shared": "3.3.4",   
 "estree-walker": "^2.0.2",   
 "magic-string": "^0.30.0"   
 }   
 },   
 "node\_modules/@vue/runtime-core": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/runtime-core/-/runtime-core-3.3.4.tgz",   
 "integrity": "sha512-R+bqxMN6pWO7zGI4OMlmvePOdP2c93GsHFM/siJI7O2nxFRzj55pLwkpCedEY+bTMgp5miZ8CxfIZo3S+gFqvA==",   
 "license": "MIT",   
 "dependencies": {   
 "@vue/reactivity": "3.3.4",   
 "@vue/shared": "3.3.4"   
 }   
 },   
 "node\_modules/@vue/runtime-dom": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/runtime-dom/-/runtime-dom-3.3.4.tgz",   
 "integrity": "sha512-Aj5bTJ3u5sFsUckRghsNjVTtxZQ1OyMWCr5dZRAPijF/0Vy4xEoRCwLyHXcj4D0UFbJ4lbx3gPTgg06K/GnPnQ==",   
 "license": "MIT",   
 "dependencies": {   
 "@vue/runtime-core": "3.3.4",   
 "@vue/shared": "3.3.4",   
 "csstype": "^3.1.1"   
 }   
 },   
 "node\_modules/@vue/server-renderer": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/server-renderer/-/server-renderer-3.3.4.tgz",   
 "integrity": "sha512-Q6jDDzR23ViIb67v+vM1Dqntu+HUexQcsWKhhQa4ARVzxOY2HbC7QRW/ggkDBd5BU+uM1sV6XOAP0b216o34JQ==",   
 "license": "MIT",   
 "dependencies": {   
 "@vue/compiler-ssr": "3.3.4",   
 "@vue/shared": "3.3.4"   
 },   
 "peerDependencies": {   
 "vue": "3.3.4"   
 }   
 },   
 "node\_modules/@vue/shared": {   
 "version": "3.3.4",   
 "resolved": "https://registry.npmmirror.com/@vue/shared/-/shared-3.3.4.tgz",   
 "integrity": "sha512-7OjdcV8vQ74eiz1TZLzZP4JwqM5fA94K6yntPS5Z25r9HDuGNzaGdgvwKYq6S+MxwF0TFRwe50fIR/MYnakdkQ=="