



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

رایانش ابری
(پاییز ۱۴۰۱)

پروژه اول:

محمد چوپان ۹۸۳۱۱۲۵

گزارش پروژه:

برای پیاده سازی این پروژه از Django rest framework استفاده شده است. در مرحله اول با توجه به طراحی که در صورت پروژه گفته شده بود یک مدل پایگاه داده به شکل زیر تشکیل شده است :

```
You, 6 hours ago | 1 author (You)
class Advertisement(models.Model):
    #create choice field for advertisement type
    You, last week | 1 author (You)
    class choices(models.TextChoices):
        accepted = "accepted"
        rejected = "rejected"
        pending = "pending"

    id=models.AutoField(primary_key=True)
    description = models.CharField(max_length=200)
    email = models.EmailField()
    category=models.CharField(max_length=50,blank=True,null=True)
    state=models.CharField(max_length=50,choices=choices.choices,default=choices.pending)
    image = models.CharField(max_length=50,blank=True,null=True)

    def __str__(self):
        return self.description
```

سپس

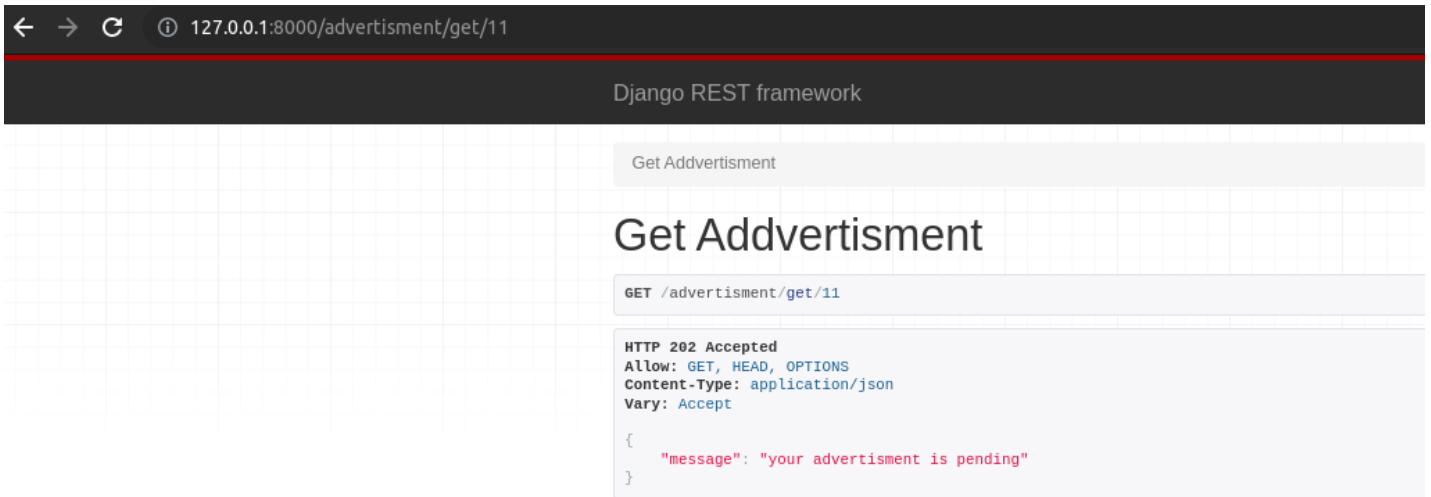
دو view برای اینکه درخواست ثبت آگهی را بررسی کند و دیگری نیز برای اینکه درخواست پیگیری را انجام دهد.

ابتدا به علت راحتی دومی که به هیچ سرویسی نیاز نداشت پیاده سازی شد به صورت زیر :

```
class GetAddvertisementView(generics.RetrieveAPIView):
    serializer_class=AdvertisementSerializer
    queryset=Advertisement.objects.all()
    def get(self,request,id):
        add=get_object_or_404(Advertisement,id=id)
        serializer=AdvertisementSerializer(add)
        print(serializer.data)
        data=serializer.data
        if(data['state']=="accepted"):
            return Response(data,status=status.HTTP_200_OK)
        elif (data['state']=="pending"):
            return Response({"message":"your advertismnt is pending"},status=status.HTTP_202_ACCEPTED)
        else:
            return Response({"message":"your advertismnt is rejected"},status=status.HTTP_403_FORBIDDEN)
```

که یک شماره آگهی را گرفته و تایید شدن یا نشدن آن را باز میگرداند. تصویر نتیجه :

که ایدی را کوئری پارام میگیرد.



برای پیاده سازی API اول نیز ابتدا باید پایگاه داده تنظیم می شد که از سایت [رانفلر](#) استفاده شده است برای اینکار تنها نیاز است که آدرس پایگاه داده ای که به ما میدهد را در این قسمت settings.py عوض کنیم.

```
# DATABASES = {  
#     'default': {  
#         'ENGINE': 'django.db.backends.postgresql',  
#         'NAME': 'tls_db',  
#         'USER': 'postgres',  
#         'PASSWORD': 'fdhdkj@33443',  
#         'HOST': 'runflare.run',  
#         'PORT': '5432',  
#     }  
# }
```

به علت هزینه ها فعلا لوکال اجرا کردیم تا زمان تحویل تعویض کنیم. برای ذخیره سازی عکس نیز از فضای ذخیره ساز شیع ابر آروان استفاده شده است. به طوری که عکس را از کاربر گرفته در صندوقچه بارگذاری کرده و لینک آن را در پایگاه داده ذخیره میکنیم.

Django REST framework

admin

Add Advertisement

OPTIONS

GET /advertisement/add

HTTP 405 Method Not Allowed
 Allow: POST, OPTIONS
 Content-Type: application/json
 Vary: Accept

```
{
  "detail": "Method \"GET\" not allowed."
}
```

Raw data

HTML form

Description
 Email
 Image Screenshot from 2022-08-09 18-23-57.png

POST

پس از ارسال اگر موفق باشد پیام زیر چاپ می شود :

```
Vary: Accept
{
  "message": "your advertisement submitted with id 22"
}
```

در غیر این صورت پیام مناسب را می دهد.
 برای پیاده سازی سرویس دوم از RabbitMQ گفته شده در صورت پروژه استفاده شده
 علاوه بر این از سایت imgga برای دسته بندی و از maigun نیز برای ارسال ایمیل استفاده شده
 است.

به صورت زیر :

```
class AddAdvertisementView(generics.CreateAPIView):
    serializer_class=AddAdvertisementSerializer
    queryset=Advertisement.objects.all()
    def post(self,request):
        #inja dadm be abr arvan
        new_data={
            "description":request.data["description"],
            "email":request.data["email"]
        }
        serializer=AdvertisementSerializer(data=new_data)
        if(serializer.is_valid()):
            serializer.save()
            data=serializer.data
            add_id=data['id']
            file=request.data['image']
            url=upload_to_server(file,add_id)
            if(url):
                add=Advertisement.objects.get(id=add_id)
                add.image=s3_url()+str(add_id)+".jpg"
                add.save()
                second_service_task.delay(add_id)
                return Response({"message": f"your advertisement submitted with id {add_id}"},status=status.HTTP_201_CREATED)
            else:
                return Response({"message": "something went wrong"},status=status.HTTP_500_INTERNAL_SERVER_ERROR)
        # print(serializer.errors)
        return Response({"message":"Please complete form correctly"},status=status.HTTP_400_BAD_REQUEST)
```

به صورتی است که برای پیاده سازی صف از celery استفاده شده است.
ابتدا عکس را ذخیره کرده و سرویس مورد نظر ره به صف اضافه میکنیم.
celery به صورت خودکار صف را خوانده و اجرا می کند که اجرای آن به صورت زیر است :

```
from django.conf import settings
IMG_KEY="acc_90ccaadf4bca7"
IMG_SECRET_KEY="daee709507bb27b5756e6dfac46b5222"
IMG_AUTH="Basic YWNjXzkwY2NhYWVmYm0YmNhNzpkYWVlNzA5NTA3YmIyN2I1NzU2ZTZkZmFjNDZiNTIyMg=="
IMG_ENDPOINT="https://api.imgur.com"
temp_data={}

def send_simple_message(message):
    return requests.post(
        "https://api.mailgun.net/v3/sandbox094d96c69abf48f980bb338921d399fd.mailgun.org/messages",
        auth=("api", "95dbfb630c67c77ea3ab2c439bc559b6-2de3d545-d58ccc46"),
        data={"from": "Excited User <mailgun@sandbox094d96c69abf48f980bb338921d399fd.mailgun.org>",
            "to": ["mohamadchoupan80@gmail.com","mohamadchoupan94@gmail.com"],
            "subject": "Advertisement APP",
            "text": message})
```

```

url=create_image_url(id)
response = requests.get(
    IMG_ENDPOINT+'v2/tags?image_url=%s' % url,
    auth=(IMG_KEY, IMG_SECRET_KEY))
data=response.json()
# we use temp data for testing
res=temp_data["result"]["tags"]
max_confidence=0
max_tag=""
for i in res:
    text=i["tag"]["en"]
    confidence=i["confidence"]
    if(text == "car" or text == "vehicle" or text == "transportation" or text == "automobile" or text == "motor vehicle" or text == "motorcar" or text == "machine" or text == "mo
        if(confidence>max_confidence):
            max_confidence=confidence
            max_tag=text
if(max_confidence>0):
    advertismment=Advertisement.objects.get(id=id)
    advertismment.state="accepted"
    advertismment.save()
    message="Advertisement with id "+str(id)+" is accepted"
    send_simple_message(message)
if(max_confidence==0):
    advertismment=Advertisement.objects.get(id=id)
    advertismment.state="rejected"
    advertismment.save()

```

به صورتی که به تابع ها به همراه آیدی که صدا زده شده اند درون صف اند و به ترتیب اجرا می شوند. که ابتدا آدرس عکس را برای سرویس دسته بندی فرستاده سپس نتایج آمده را تحلیل کرده و در نهایت نیز با توجه به تحلیل ایمیل را به کاربر زده و وضعیت آگهی را عوض میکنیم به جز پایگاه داده تمامی سرویس ها موارد توصیه شده در صورت پروژه بود و همه یک API key داده و با توجه به آن احراز هویت میکردند.

که در قسمت ابر آروان باید دسترسی عمومی فایل را می دادیم تا بتوان آن را تحلیل کرد در غیر این صورت باید شرایط پیچیده تر می شد.