



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

گزارش تمرین اول

درس رایانش ابری

هلیا سادات هاشمی پور - ۹۸۳۱۱۰۶

استاد

دکتر جوادی

آبان ۰۱

ابتدا سرویس های ابری استفاده شده را نام میبرم:

- برای میزبان ابری از ابرآروان استفاده شده است.
- برای پایگاه داده از Aiven استفاده کردم.(mySQL)
- سایر سرویس ها هم طبق دستور کار استفاده شد. مثل RabbitMQ, Imagga و mailgun
- برای ذخیره ساز شی هم از ابرآروان استفاده شد که در آن یک صندوقچه ساختم.

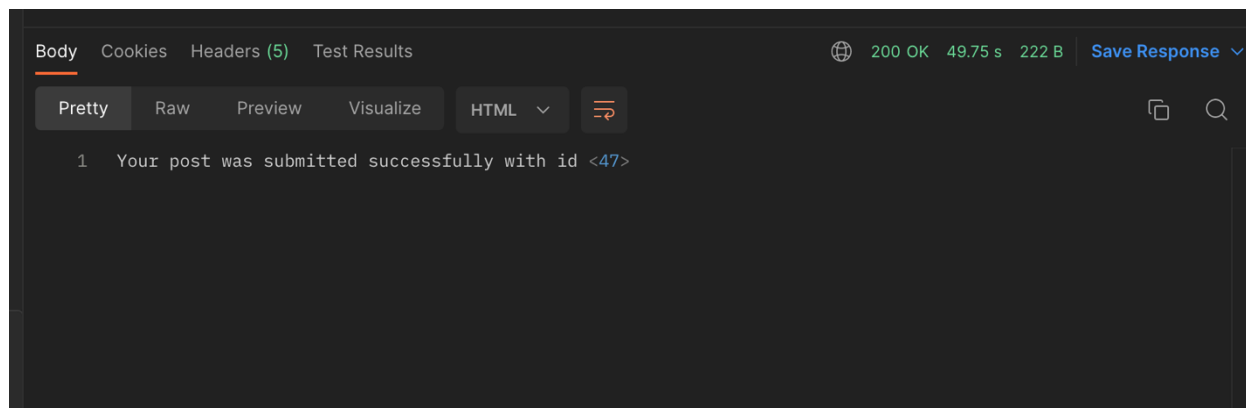
حال به توضیح مختصری درمورد پروژه میپردازم

همانطور که در صورت پروژه هم آمد، نرم افزار ما از دو سرویس بکند(قسمت بکند هم با flask زده شده است) تشکیل میشود. سرویس اول وظیفه دریافت درخواست کاربران و پاسخگویی به آن را دارد. سرویس دوم وظیفه پردازش (تعیین دسته بندی و یا رد آگهی) را دارد .

طبق مواردی که در دستور کار گفته شد برای هر دو سرویس گام به گام عمل کردیم. برای تست هم از postman استفاده کردم. علاوه بر شمای پیشنهادی که در دستور کار ذکر شد، یک ستون image_type برای نوع فایل عکس در دیتابیس قرار دادم. کدهای آماده برای هر سرویس هم در اختیارمان بود فقط برای به دیتابیس توابع بیشتری اعم از update اضافه گردید.

در اصل برای سرویس اول و بخش ثبت آگهی کاری که کردم این بود که اول API اطلاعات یک آگهی شامل متن، تصویر و ایمیل فرستنده بود را دریافت میکند سپس اطلاعات این آگهی، شامل متن و آدرس ایمیل فرستنده را در پایگاه داده توسط تابع insert ذخیره میکنیم و برای آن یک id در نظر میگیرد.(در کد هر بخش هم به شکل کامنت توضیح داده شده) حال این API اطلاعات ما یک آگهی شامل متن، تصویر و ایمیل فرستنده آن را دریافت کرده و تصویر آن را در یک ذخیره ساز شی ذخیره کردم. سپس هم id آگهی را برای پردازش داخل صف RabbitMQ نوشتم.

در آخر هم پیام Your post was submitted successfully with id را برگرداندم.(عکس را از لوکال خودمان گرفتیم که کد اینکه از طریق url هم بتوان دریافت کرد را کامنت کردم)



حال برای دریافت آگهی هم این API، id یک آگهی را دریافت کرده و براساس وضعیتش پیغام مرتبط داده میشود اگر آگهی ما تایید شده باشد به معنای آنکه vehicle هست (is_vehicle=true) اطلاعات این آگهی شامل متن، تصویر و دسته‌بندی و وضعیت آن را در پاسخ برمیگردانند.(تمامی در بکند بذرنامه نوشته شده هندل شده است)

برای سرویس دوم هم کد RabbitMQ_recieve را در تابع callback تغییر ایجاد کردم. به این شکل که در هر پیام خوانده شده از صف، یک id آگهی وجود دارد. با این id، عکس آگهی از ذخیره‌سازی دریافت میشود. عکس آگهی، به سرویس imagga عکس برای پردازش ارسال میشود. از پاسخ سرویس imagga، اولین برچسب برای دسته‌بندی آگهی انتخاب میکنیم. (برای تست هم عکس با کیفیت باید داده شود) این دسته‌بندی را در ستون category دیتابیس قرار دادم. در اصل از تابع update برای به روزرسانی اطلاعاتم اعم از استیت و category استفاده کردم. با استفاده از سرویس mailgun، ایمیلی به اکانت خودم ارسال کردم تا از وضعیت (تایید یا رد شدن) آگهی خود مطلع شوم.

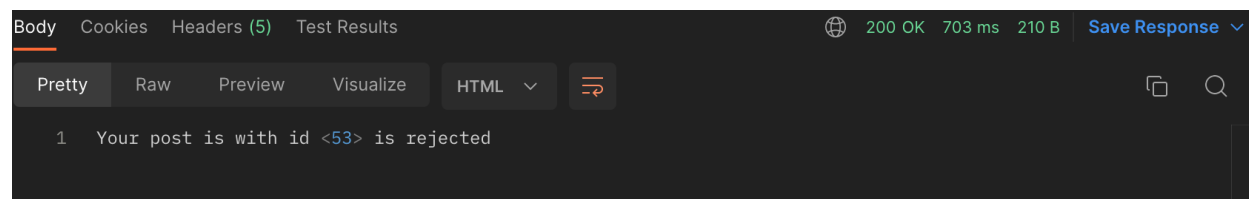
زمانیکه آگهی تایید شده باشد و یک اتومبیل داشته باشیم اگر شناسه را بخواهیم دیتای مورد نیاز و تصویر response داده میشود. اگر دقت شود تصویر را encode کردن به base64 و در postman عکس رو نشان نمیدهد و base64 را میدهد که میتوان آنرا با استفاده از conventorهای آنلاین به عکس تبدیل کرد.

```
Body Cookies Headers (5) Test Results 200 OK 3.73 s 44.29 KB Save Response
Pretty Raw Preview Visualize JSON
1
2 "data": [
3   "description: This is the car",
4   "category: car",
5   "state: 2"
6 ],
7 "image": "/9j/4AAQSkZJRgABAQAAQABAAD//
  gA7Q1JFQVRPUjogZ2QtanBlZyB2MS4wICh1c2luZyBJSkcgS1BFYyB2NjIpLCBxdWFSaXR5ID0gOTUK/
  9sAQwACAQEBAQECAQEBAgICAgIEAwICAgIFBAQDBAYFBgYGBQYGBGcJCAyHQCcGBggLCAkKCgoKCgYICwwLCgwJCgoK/
  9sAQwECAgICAgIFAwMFCgcGBwoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoK/
  8IAEQgBLAHCawEiAAIRAQMRAf/EABwAAAEFAQEBAQAAAAAAAAAQCAwUGBwABCP/
  EABoBAAMBAAQEBAQAAAAAAAAAAAwEEBQb/2gAMAwEAAhADEAAAAf1stk/1vLcMbKAYSh3kLJI1Y
  +VFQswLpGpkedQCihFojuR41J69BwPd4y013Dp6I8cVKkYTI0z2LYJ9YimbP2EOZIVq7J4YEdsw0XI465L0
  +FixnkR6I5Ri9QHpfKwKEkRryADkueQMZah12quSRLJDyJr0yvRt4GFovW/qy+bD/
  DPe5HcfDcEcT2Azq0q3nqnJspbz4wzr651ZWvwZ7vXTUPNcB5MSmSDoxs9km
  +chSPMJfBCFsjiYajGUxqKRSdm1qc5G17XR0la5AVwgSbw0ctGvGgmQxmEZec/I1uTRp6Pa8k/
  owYVUm0QvC2FAU3IjenuVfmckZ76bzzC4t1Cb9jT8ZbjXpR1xkhQk0YybFo9enUMaX51h1yvof7KLKwvS7QMgJpYkkuHJi8i4
  BIozKZF6bQ8XZhmWpB2GC6pRLRn1JelrfnWRuNjtvB1T3M9x9kbWLNc9XLAdLeVhGuF8AzR3BF9KuhBKPBFJkoy1yaT6T7jt8k
  kNPFv+Kpzxw3jBnsY58M5K
  +8WYNHGFkzZEQIQuypIBmqNLJJXXeaQjBIKegsZ0w0EUxOMmRkywSpJE9YuW8Uu0rhX0ArtCdEM9h7cL6HHEFkp1VUqCqFT3IF
  c2nnRCUqlsv1QNs0VsEYkn9WvItryFaesTYQlkcMhvcz0W+YyNJ8
  +n8f0yLTFUwtrSL2kCuxctYclSxniRFRob4khWHLkEKYjAZFNeKkUS0EL17cMdam5VbesMtx9dGGv1crGNkmDXR4kuW571QDUY
```

در فایل code2 صرفا کاری که کردم این بود که هم به شکل url عکس را میگیرتم و در خروجی هم response ای که میگیریم عکس به شکل url است(همان url عکسمان در صندوقچه ابرآوان است) و کد app.py و Proxies.py مقداری تغییر کرده است.

```
1
2 "data": [
3   "description: This is the car",
4   "category: car",
5   "state: 2"
6 ],
7 "image_url": "https://hw1cloudcomputing.s3.ir-thr-at1.arvanstorage.com/hw1cloudcomputing/52.jpg?
  AWSAccessKeyId=6d04e8b0-645e-4a26-b9e0-128f830a36ce&Signature=viMwmuxHiDNyWVu00FCh7KtG3UY%3D&
  Expires=1668586113"
8
```

زمانی که آگهی تایید شده نباشد(عکس یک توربین را دادم) response ی که داریم به شکل زیر است.



ایمیل دریافتی هم به صورت زیر میباشد.

