به نام خدا

دانشگاه صنعتی امیرکبیر (پلی‌تکنیک تهران)

تمرین۱: سامانه ثبت آگهی

درس:

رایانش ابری

دانشجو:

امیرحسین علی‌بخشی

شماره دانشجویی:

۹۷۳۱۰۹۶

استاد:

دکتر جوادی

پاییز ۱۴۰۱

فهرست مطالب

۱ ساختار پروژه

۱ - ۱ فریمورک استفاده شده برای نوشتن APIها

برای انجام این پروژه، از فریم‌ورک اکسپرس (express) که به زبان برنامه‌نویسی جاوااسکریپت می‌باشد استفاده شده است. به کمک این فریم‌ورک می‌توان به راحت APIهای دلخواه را ایجاد کرده و از آن‌ها استفاده کرد.

۱ - ۱ - ۱ لیست APIها

از APIهای موجود در این برنامه می‌توان به موارد زیر اشاره نمود. (توضیحات کامل‌تر داخل فایل readme قرار دارند)

ردیف	متد	آدرس	توضیحات
۱	GET	/ad	دریافت لیست تمام آگهی‌ها
۲	GET	/ad/:id	دریافت اطلاعات یک آگهی با استفاده از شناسه‌ی تبلیغ
۳	POST	/ad	ثبت یک آگهی جدید (به کمک عکس، آدرس ایمیل و توضیحات آگهی)
۴	DELETE	/ad	حذف یک آگهی
۵	DELETE	/ad/:id	حذف تمام آگهی‌ها

۱ - ۲ سرویس‌های استفاده شده

نام سرویس	توضیحات
پایگاه داده	برای این پروژه از دیتابیس MongoDB استفاده شده است که برای استفاده از آن از سرویس Mongo Atlas بهره می‌بریم.
آبجکت استورج	برای ذخیره‌سازی عکس‌ها از سرویس S3 شرکت ابرآروان استفاده شده است.
سایر موارد	سایر سرویس‌ها همان سرویس‌هایی می‌باشند که در دستور کار به آن‌ها اشاره شده است.

۱ - ۳ پوشه‌بندی پروژه

پروژه دارای چندین پوشه اصلی است که شرح هریک به صورت زیر می‌باشد:

نام پوشه	توضیحات
dataaccess	توابع لازم برای کار با پایگاه داده
DTO	فیلتر کردن فیلدهای موجود در اسناد داخل پایگاه‌داده (Data Transfer Object)
models	تعیین کردن مدل و اسکیمای مروبط به پایگاه‌داده
server	کد مربوط به سرورهای A و B در این پوشه قرار دارد. در واقع APIهای ما در این پوشه نوشته شده است.
services	این بخش‌شامل فایل‌های مربوط به سرویس‌های استفاده شده در این پروژه (بجز MongoDB) می‌باشد.\
utils	توابع کاربردی استفاده شده در کل پروژه

۲ اجرای پروژه

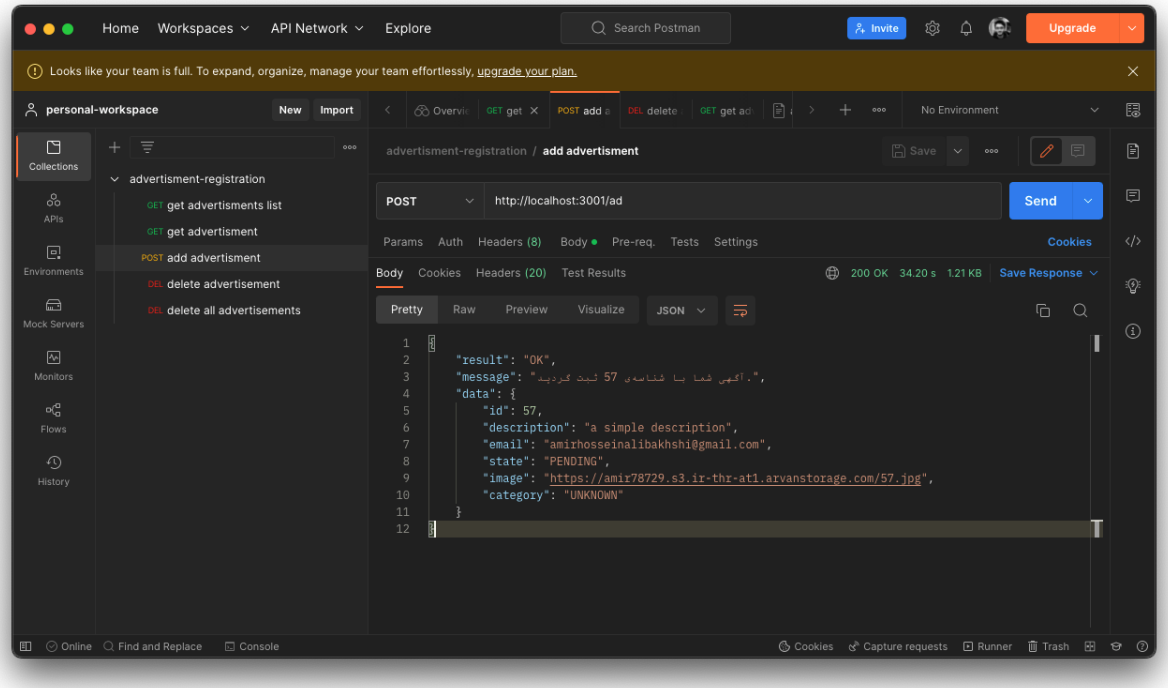
برای اجرای پروژه باید دستور زیر را اجرا کرد:

```
node src
```

به کمک این دستور، فایل src/index.js اجرا می‌شود که در آن سرور A روی پورت ۳۰۰۱ و سرور B روی پورت ۳۰۰۲ به حالت اجرا در می‌آید و سرویس آماده‌ی استفاده می‌گردد.

```
[ServerA] listening on port 3001
[ServerB] listening on port 3002
[ServerB/RabbitMQ] connecting to RabbitMQ...
[ServerB/RabbitMQ] connected to RabbitMQ on queue "amir78729"
[MongoDB] connection established successfully
```

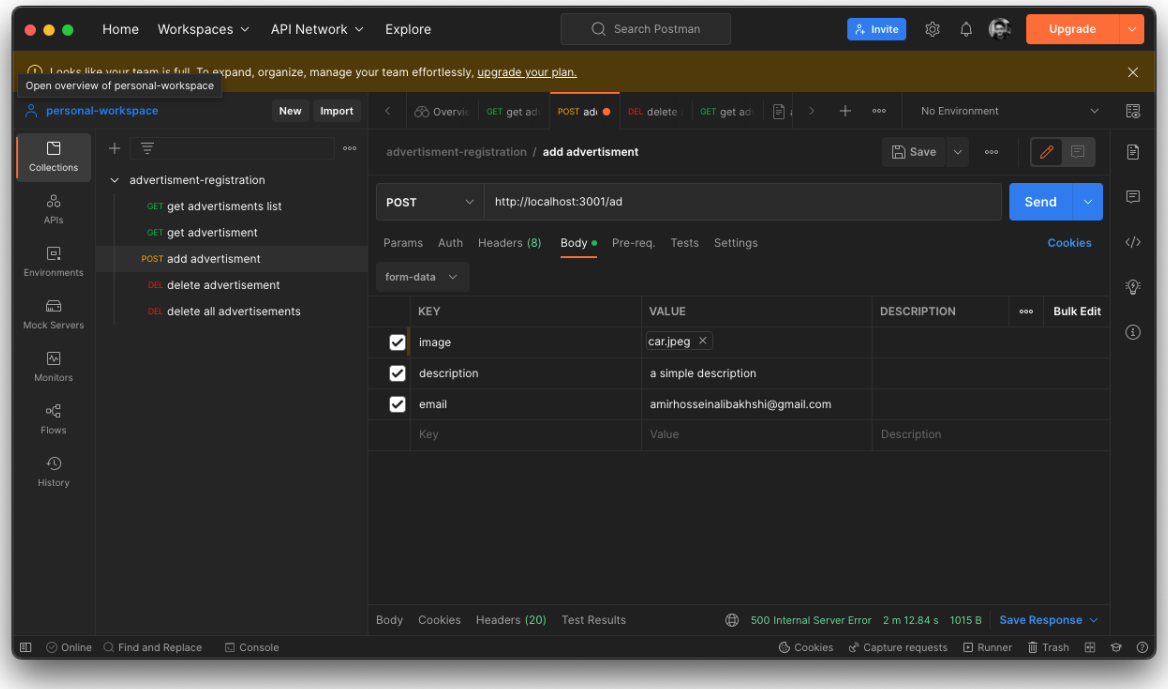
به کمک ابزارهای مختلفی می‌توان API call انجام داد. ما در این پروژه از ابزار postman استفاده می‌کنیم. لیست APIهای این پروژه به صورت زیر می‌باشد:



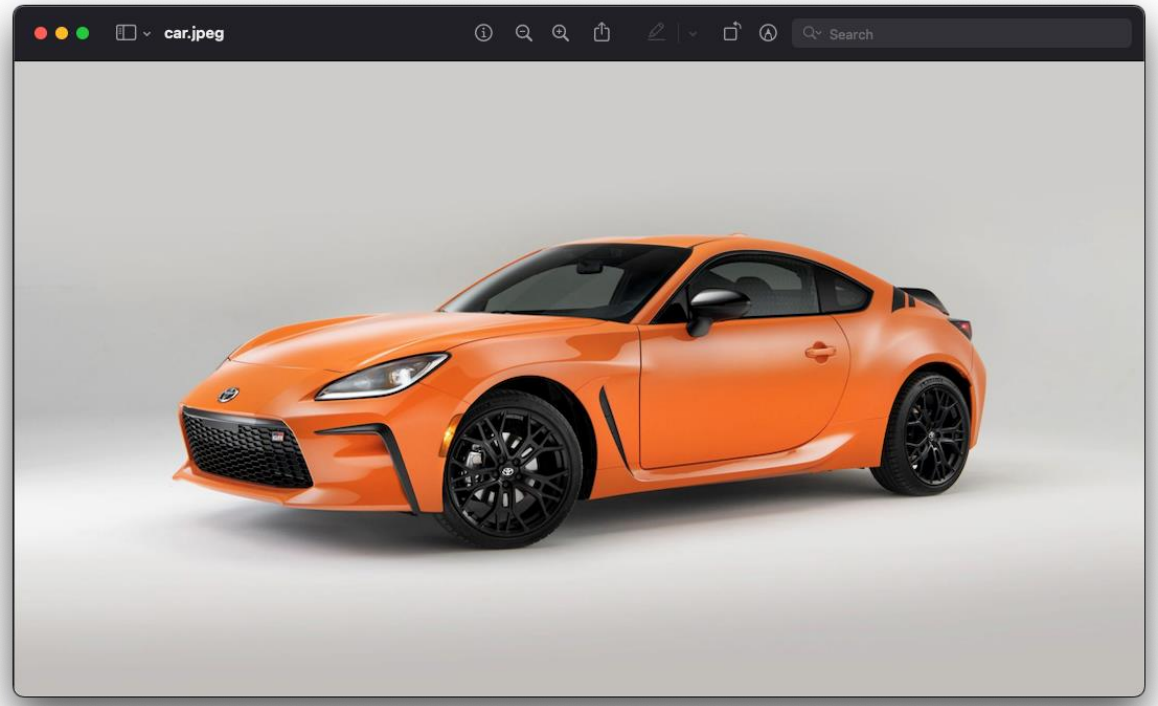
۲ - ۱ تشریح فرآیند ثبت آگهی

۲ - ۱ - ۱ بخش اول: سرور A

برای ثبت آگهی، کلایت از طریق postman باید API ثبت آگهی (با متد POST) را فراخوانی کند. داخل این درخواست باید اطلاعات مربوط به آگهی را به صورت Form Data در body این درخواست قرار دهیم.



در این مثال ما عکس خودروی زیر را برای سرور ارسال خواهیم کرد:

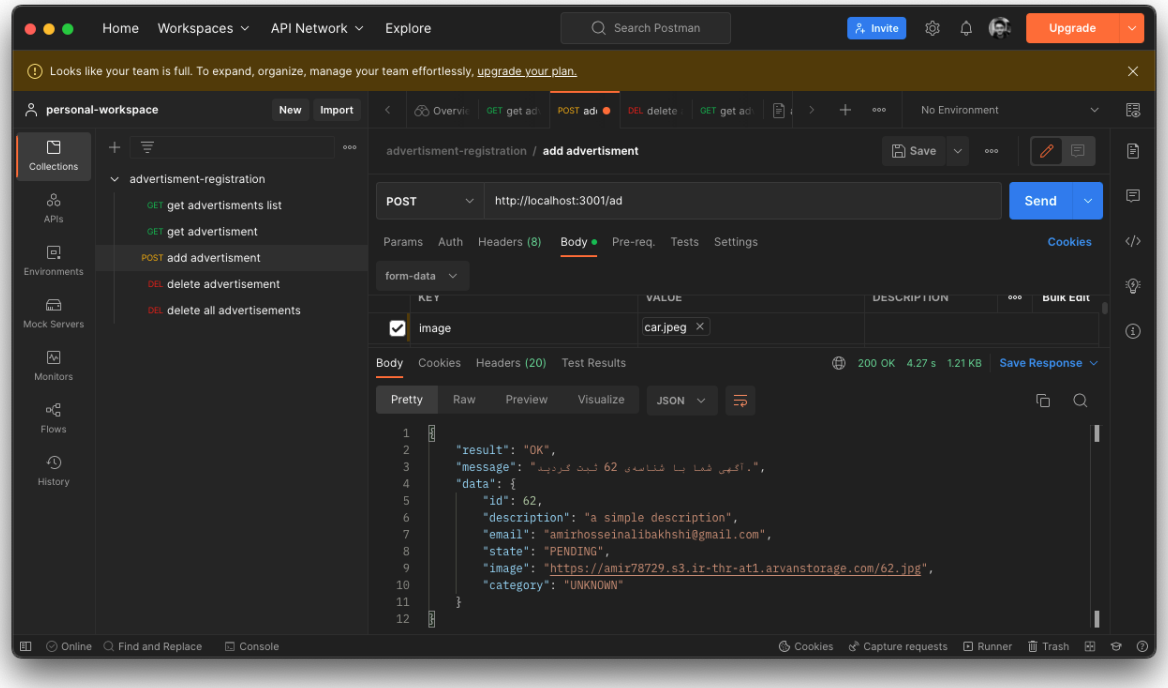


با ارسال این درخواست، سرور A با توجه به آگهی‌های موجود در پایگاه‌داده، شناسه‌ی این درخواست را محاسبه می‌کند. سپس اطلاعات آگهی را به پایگاه‌داده ارسال می‌کند. پس از ثبت اطلاعات در پایگاه داده، نوبت به آپلود عکس در S3 می‌رسد که این کار بسته به حجم عکس، می‌تواند اندکی زمان‌بر باشد. نام عکس آپلود شده همان شناسه آگهی می‌باشد که دارای فرمت jpg می‌باشد.

آخرین کار، ارسال شناسه آگهی جدید به RabbitMQ می‌باشد. سرور ابتدا اتصال خود را با این سرویس برابر می‌کند و سپس شناسه را به آن ارسال می‌کند. تمامی مراحل گفته شده داخل کنسول سرور قابل مشاهده است.

```
[MongoDB] finding last advertisement's id...
[MongoDB] adding new advertisement...
[ServerA/S3] sending file to s3...
[ServerA/S3] file was sent successfully {
  '$metadata': {
    httpStatusCode: 200,
    requestId: 'tx000005a1baa503427cfcb-00637636c7-294974a1-ir-thr-at1',
    extendedRequestId: undefined,
    cfId: undefined,
    attempts: 1,
    totalRetryDelay: 0
  },
  ETag: '"46dcbae37309b698c28da54c9b88eee9"'
}
[ServerA/RabbitMQ] connecting to RabbitMQ...
[ServerA/RabbitMQ] connected to RabbitMQ on queue "amir78729"
[ServerA/RabbitMQ] message "62" was sent to queue "amir78729"
```

حال در پاسخ درخواست کلاینت، پیام زیر برای او ارسال می‌گردد:

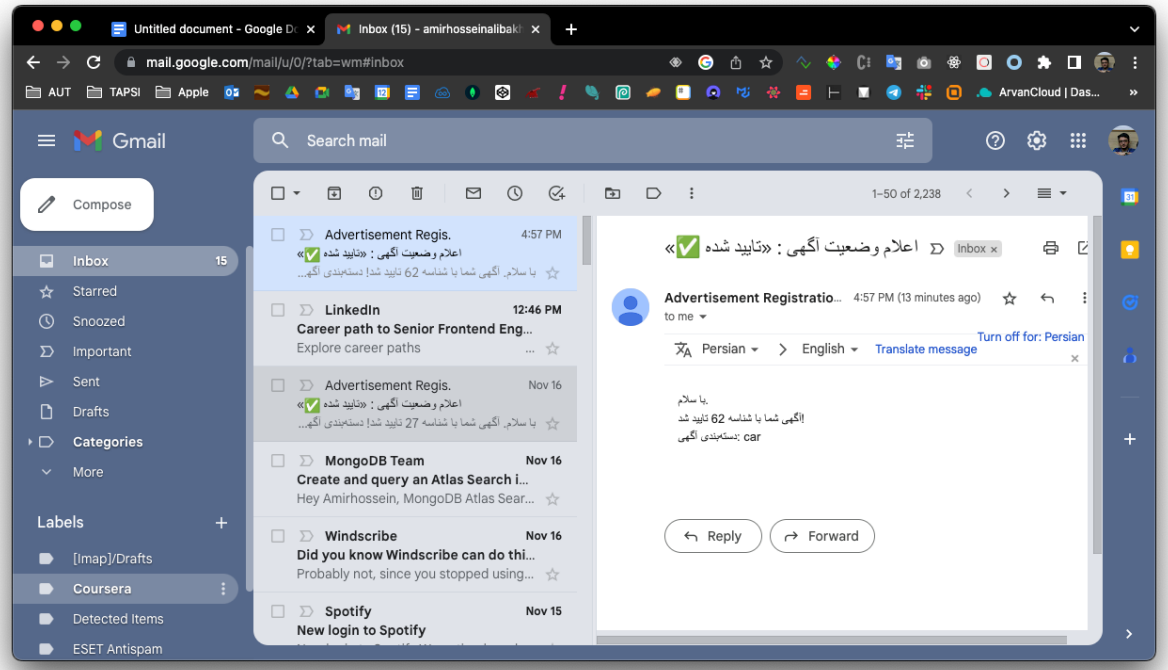


۲ - ۱ - ۲ بخش دوم: سروس B

از آنجا که سرور B مدام در حال گوش دادن به صف تعریف شده در سرویس RabbitMQ می‌باشد، زمانی که سرور A شناسه آگهی را داخل صف فرستاد پس از زمان کمی این شناسه در سرور B قابل دسترسی می‌باشد. بخاطر این که نام هر عکس داخل S3 برابر با شناسه آگهی می‌باشد، آدرس URL عکس مربوط به این آگهی به سرویس Imagga ارسال می‌گردد. پس از مشخص شدن نتایج، اطلاعات این آگهی داخل پایگاه داده به‌روزرسانی می‌شود و در پایان نتیجه این عملیات به ایمیل کاربر ارسال می‌گردد. همانند سرور A، تمام مراحل در سرور B نیز در کنسول قابل مشاهده هستند.

```
[ServerB/RabbitMQ] received message: 62
[ServerB/RabbitMQ] processing image for ad 62...
[ServerB/RabbitMQ] results for ad 62: state: APPROVED, category: car
[MongoDB] update add with id = 62...
[MongoDB] ad 62 status was update
[ServerB/MailGun] sending advertisement status to user's email...
[ServerB/MailGun] mail was sent successfully
{
  id: '<20221117132749.827953b3ec546cf4@sandboxa7eb94664947415982636a4b9fd2bfdd.mailgun.org>',
  message: 'Queued. Thank you.'
}
```

ایمیل ارسال شده به کاربر به صورت زیر می‌باشد:



حال با کمک APIهای تغریف شده برای مشاهده وضعیت آگهی‌ها، این آگهی را مشاهده کرد: