

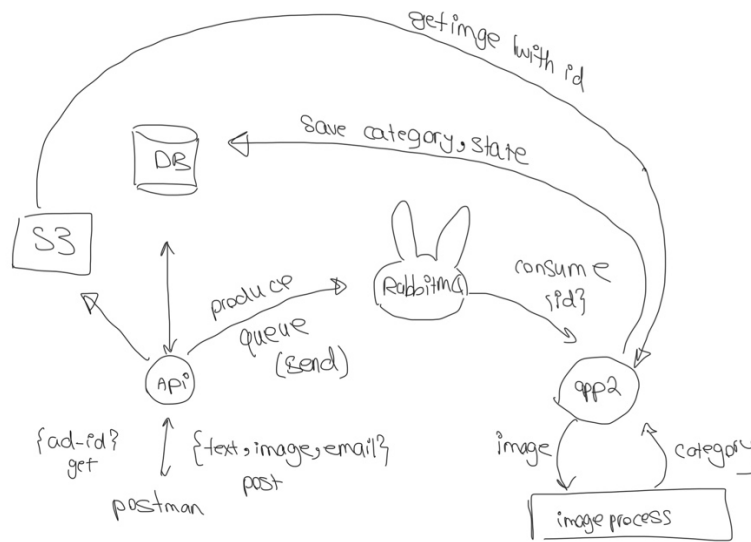
گزارش تمرین اول

تاریخ: ۱۴۰۱/۸/۱۷

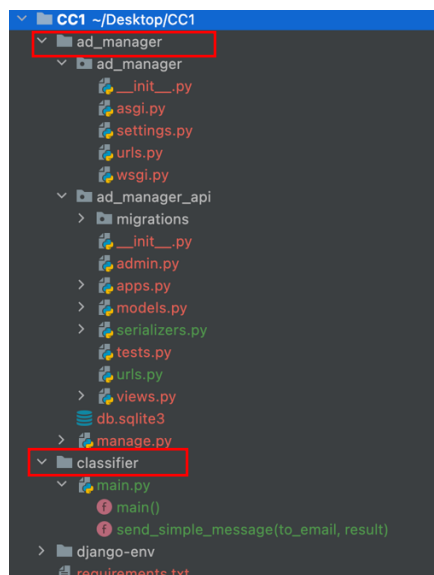
شماره دانشجویی: ۹۸۳۱۱۱۸

نام و نام خانوادگی: روژینا کاشفی

شمای کلی پروژه به صورت زیر است:



ساختار پوشه‌بندی پروژه به صورت روبه‌رو است. شامل دو فولدر اصلی است که هر کدام یک سرویس هستند.



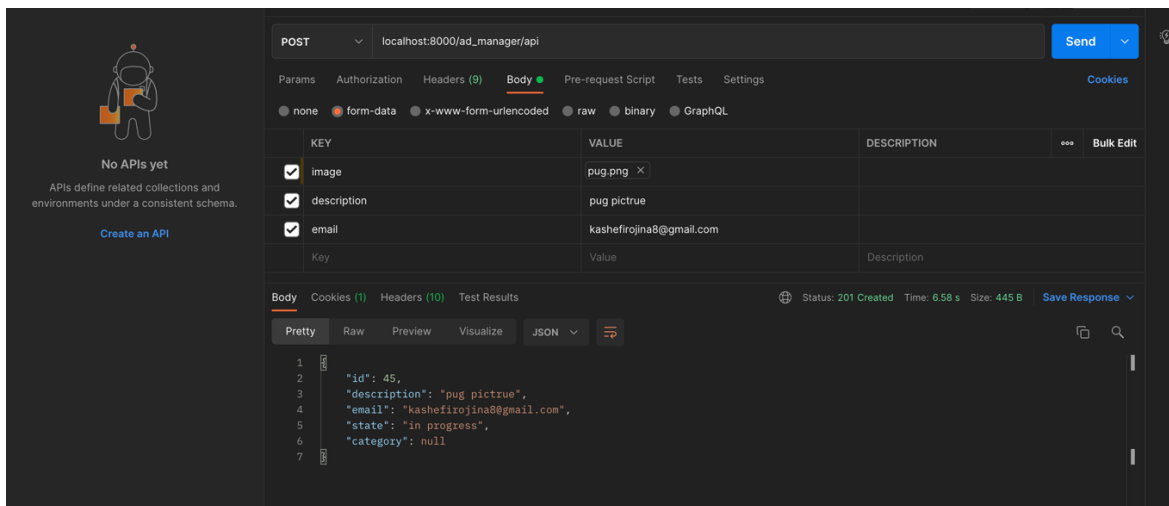
• سرویس اول (ad manager)

شامل دو پوشه است. پوشه ad_manager محتوایی است که خود Django برای مدیریت پروژه درست می‌کند و ما در setting.py تغییرات سیستمی اعمال می‌کنیم. اطلاعات سرویس ابری avien که برای دیتابیس postgres استفاده شده و باکت s3 در این قسمت وارد شده است. برای سرویس s3 از aws استفاده شده است. پوشه دوم ad_manager_api است. در فایل model مدل و نحوه ذخیره‌سازی اطلاعات در پایگاه داده ریموت نوشته شده و سپس در فایل view دو api یکی post برای ثبت آگهی و دیگری get برای دریافت وضعیت آگهی نوشته شده است. هنگام post باید ایمیل و توضیحات در دیتابیس و تصویر در باکت s3 ذخیره و برای RabbitMQ که از سرویس cloudAMQP استفاده شده، ارسال شود.

• سرویس دوم (classifier)

این سرویس وظیفه خواندن شناسه عکس مورد بررسی از صف را دارد که و سپس عکس مورد نظر را با استفاده شناسه از باکت s3 می‌خواند و با استفاده از سرویس مشخص کننده زمینه عکس state و category عکس را شناسایی و در دیتابیس اپدیت می‌کند.

برای کارکردن با سرویس RabbitMQ از سرویس ابری cloudAMQP استفاده می‌کنیم و برای تگ زدن عکس‌ها جای پیاده‌سازی یک classifier از سرویس Imagga استفاده می‌کنیم. همانطور که در تصاویر زیر مشاهده می‌کنید با استفاده از متد post کاربر می‌تواند آگهی ثبت کند و با استفاده از متد get می‌تواند وضعیت آگهی را مشاهده کند.



GET localhost:8000/ad_manager/api Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> id	32			
Key	Value	Description		

Body Cookies (1) Headers (10) Test Results Status: 200 OK Time: 3.95 s Size: 375 B Save Response

Pretty Raw Preview Visualize

```
{ "IN PROGRESS": "Your ad in still in progress" }
```

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> id	49			
Key	Value	Description		

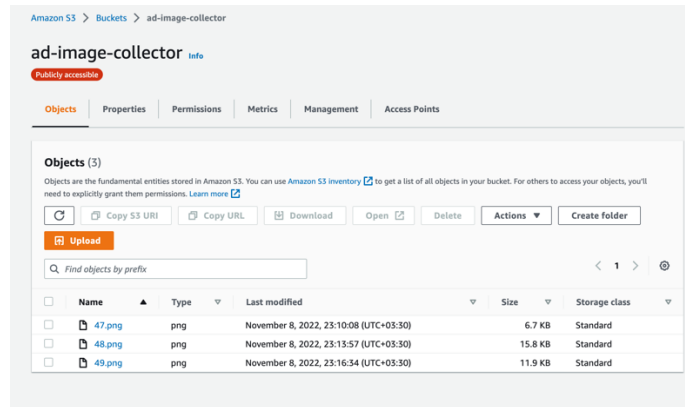
Body Cookies (1) Headers (10) Test Results Status: 200 OK Time: 1806 ms Size: 516 B Save Response

Pretty Raw Preview Visualize JSON Copy Search

```
1 200
2 "SUCCESS ": "Your ad with ID : 49 is ACCEPTED, \n with DESCRIPTION : car pictrue, \n with CATEGORY : jeep, \n with IMAGE :
3 https://ad-image-collector.s3.us-east-2.amazonaws.com/49.png "
```

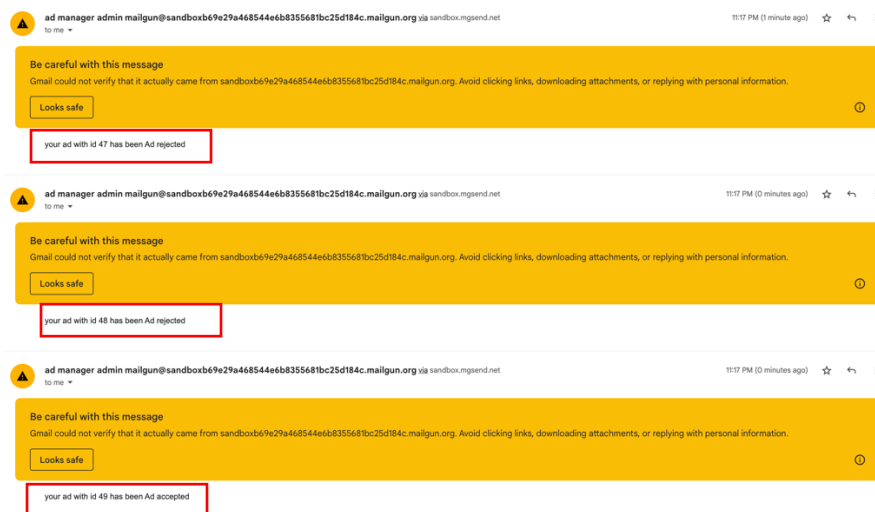
خروجی دیتابیس و 3قبل از اتصال سرویس دوم و زمانی که کاربر تازه درخواست ثبت آگهی ارسال کرده است.

```
defaultdb=> select * from ad_manager_api_ad;
id | description | email | state | category
---+-----+-----+-----+-----
47 | pug pictrue | kashefirojina8@gmail.com | in progress |
48 | sunflower pictrue | kashefirojina8@gmail.com | in progress |
49 | car pictrue | kashefirojina8@gmail.com | in progress |
(3 rows)
```



بعد از اتصال سرویس دوم تمامی موارد از صف خوانده شده و وضعیت و دسته‌بندی در دیتابیس اپدیت و ایمیل به کاربر ارسال می‌شود.

```
defaultdb=> select * from ad_manager_api_ad;
id | description | email | state | category
---+-----+-----+-----+-----
47 | pug picture | kashefirojina8@gmail.com | ad rejected | 
48 | sunflower picture | kashefirojina8@gmail.com | ad rejected | 
49 | car picture | kashefirojina8@gmail.com | ad accepted | jeep
(3 rows)
```



سرویس‌های ابری استفاده شده:

- Aiven پایگاه داده مجازی و استفاده از postgres استفاده شده است.
- ذخیره‌سازی شی s3 از سرویس ابری aws استفاده شده است.
- سرویس cloudAMQP برای کار کردن با سرویس RabbitMQ.
- برای تگ و شناسایی عکس‌ها از سرویس imagga استفاده شده است.
- برای ارسال ایمیل به صورت اتوماتیک از mailgun استفاده شده است.